

RFutils user guide

RFutils is a **browser-based toolkit for radio-mic frequency coordination**. It replaces three separate tools with one app: file conversion between Shure and Sennheiser ecosystems, Ofcom PMSE licence import, live receiver monitoring, intermodulation-clean coordination, allocation, and programming receivers.

Before you rely on this for a show: most of the file formats RFutils reads and writes are *reverse-engineered from real exports*, because neither Shure nor Sennheiser publishes full specifications for them. **Open every export in your own copy of Wireless Workbench or WSM and check it before the show.** Treat anything marked experimental as exactly that.

The tabs

Tab	What it's for	Needs the local app
Convert › Coordination files	Move coordination data between Shure Wireless Workbench (WWB) and Sennheiser Wireless Systems Manager (WSM), or in and out of plain CSV	no
Convert › Ofcom PMSE licence	Turn an Ofcom PMSE licence schedule PDF into importable frequency data	no
Inventory	Keep a list of the receivers and transmitters you own	no
Coordination	Calculate an intermodulation-clean set of frequencies	no
Allocation	Work out which frequencies go to which channel	no
Monitor	Discover Shure/Sennheiser/AES67 receivers and watch audio, battery and RF	yes
Deployment	Push assignments out to real receivers	yes

Monitor and Deployment need sockets a browser tab cannot open, so they only appear in the desktop/self-hosted build. The hosted version at rfutils.stoatworks-labs.com runs everything else entirely on your own machine — nothing you load is uploaded anywhere.

Running it

As a desktop app

RFutils ships as a tray application (built on av-launcher, with a Node runtime embedded). Start it, pick a network interface and port, press **Start**, then **Open**.

On macOS, if the app opens but the server never starts: this is almost always Gatekeeper. For an unsigned app bundling helper binaries, approving the *app* does not unquarantine the *helpers* — they are terminated silently, with no error shown. See the release notes for the unquarantine step.

From source

```
npm install
npm run dev
```

Then open **http://localhost:8420**.

No hardware? Use demo mode:

```
npm run dev:demo
```

This runs with simulated receivers, so you can explore discovery, monitoring and the whole UI without a single real device on the network. It is also the fastest way to tell an app problem from a network problem.

Converting a coordination file

1. Go to **Convert > Coordination files**.
2. Drop in your `.wwb`, WSM export or CSV. The format is detected automatically.
3. For a plain CSV, RFutils reads the header and suggests which column is which. **Check the suggested mapping** — it is a best guess from the column names.
4. Choose your export format and download.
5. **Open the result in WWB or WSM and confirm it looks right** before using it.

Coordination files (WSM · WWB · CSV) Ofcom PMSE licence (PDF)

Drop a WWB / WSM export or CSV here, or click to choose

Detected WSM Coordination Report (HTML) — 11 channel(s).

Export as

WWB frequency list (.txt)

Download

Shure's documented 'Import Frequencies from File' format — the safe option.

Name	Freq (MHz)	Zone	Group	Ch	Type	Manufacturer	Notes
EM3732	470.500	FM Mics			EM 3731/3732	Sennheiser	TX: SK5212/SKM5200, squelch 0 µV
EM3732	471.700	FM Mics			EM 3731/3732	Sennheiser	TX: SK5212/SKM5200, squelch 0 µV
SR2000 I	556.600	IEM Systems			SR 2000/2050-IEM	Sennheiser	TX: EK 2000-IEM, squelch -1 dB
SRG4	557.600	IEM Systems			SR IEM G4	Sennheiser	TX: EK IEM G4, squelch -1 dB
SRG4	556.000	IEM Systems			SR IEM G4	Sennheiser	TX: EK IEM G4, squelch -1 dB
SRG4	557.000	IEM Systems			SR IEM G4	Sennheiser	TX: EK IEM G4, squelch -1 dB
SRG4	558.000	IEM Systems			SR IEM G4	Sennheiser	TX: EK IEM G4, squelch -1 dB
EW-DX 1	470.200	Digital devices			EW-DX EM2/EM4 (LD off)	Sennheiser	TX: EW-DX SKM, squelch -87 dBm
EW-DX 2	470.800	Digital devices			EW-DX EM2/EM4 (LD off)	Sennheiser	TX: EW-DX SKM, squelch -87 dBm
EW-DX 3	471.400	Digital devices			EW-DX EM2/EM4 (LD off)	Sennheiser	TX: EW-DX SKM, squelch -87 dBm
EW-DX 4	472.000	Digital devices			EW-DX EM2/EM4 (LD off)	Sennheiser	TX: EW-DX SKM, squelch -87 dBm

Importing an Ofcom PMSE licence

1. Go to **Convert > Ofcom PMSE licence**.
2. Upload the licence schedule PDF exactly as Ofcom issued it.
3. If you get *"not a recognisable PMSE licence schedule"*, the file was readable but isn't the expected document — check you have uploaded the schedule itself rather than a covering letter or a re-saved copy.
4. Export to your coordination software.

RFutils

RF coordination & wireless-mic suite

Convert

Monitor

Coordination

Allocation

Deployment

Coordination files (WSM · WWB · CSV)

Ofcom PMSE licence (PDF)

Drop an Ofcom PMSE licence schedule PDF here, or click to choose

Parsed 6 frequency assignment(s).

Licence No.

7/9999999

Licensee

Example Events Ltd

Address

12 Demo Road, Exampleton, EX1 2MP

Period

01 Aug 2026 – 31 Aug 2026

PMSE ref.

J999999

WWB frequency list (.txt)

Reference sheet (.csv)

WWB7 show file (.shw) — experimental

The frequency list is the safe file to import directly into Wireless Workbench (Import > frequencies from file). The reference sheet maps each frequency to a suggested name and its coordination group.

Freq (MHz)	Equipment	Model	Group	Site
606.100	Wireless Microphone	G56	26NW1*6	Demo Arena, Example Town EX1 2MP
607.700	Wireless Microphone	G56	26NW1*1	Demo Arena, Example Town EX1 2MP
613.300	Wireless Microphone	G56	27NB1*1	Demo Arena, Example Town EX1 2MP
622.500	Wireless Microphone	G56	27NB1*1	Demo Arena, Example Town EX1 2MP
631.900	Wireless Microphone	G56	30NW1*1	Demo Arena, Example Town EX1 2MP
633.100	Wireless Microphone	G56	30NW1*1	Demo Arena, Example Town EX1 2MP

AI-assisted build · combines wsm-wwb-bridge, pmse-to-wwb and MicWizard · verify experimental exports before a live show.

This parser **has been validated against a real Ofcom licence**, which is more than can be said for most of the formats here.

Coordination

Computes a set of mutually-compatible frequencies — spaced, clear of excluded spectrum, and free of third and fifth-order intermodulation products.

RFutils

RF coordination & wireless-mic suite

Convert

Monitor

Inventory

Coordination

Allocation

Deployment

Compute a set of mutually-compatible frequencies — spaced, clear of excluded spectrum, and free of third/fifth-order intermodulation products.

Equipment profile

Sennheiser EW-DX

Band preset

UK UHF PMSE (470–608) · UK

Sennheiser EW-DX: ~200 kHz occupied, 0.35 MHz spacing recommended. Monitored over SSC. Verify against your unit.

Tuning ranges — MHz, e.g. 606.5–614, 470–478

470–608

How many

8

Min spacing (MHz)

0.35

Grid step (MHz)

0.025

IM guard (MHz)

0.05

Exclusion guard (MHz)

0.1

Exclusions — MHz ranges, e.g. 500–510, 520.5–521

Locked / existing (MHz, comma-separated)

☒ Avoid 3rd-order IM

☐ Avoid 5th-order IM

Coordinate

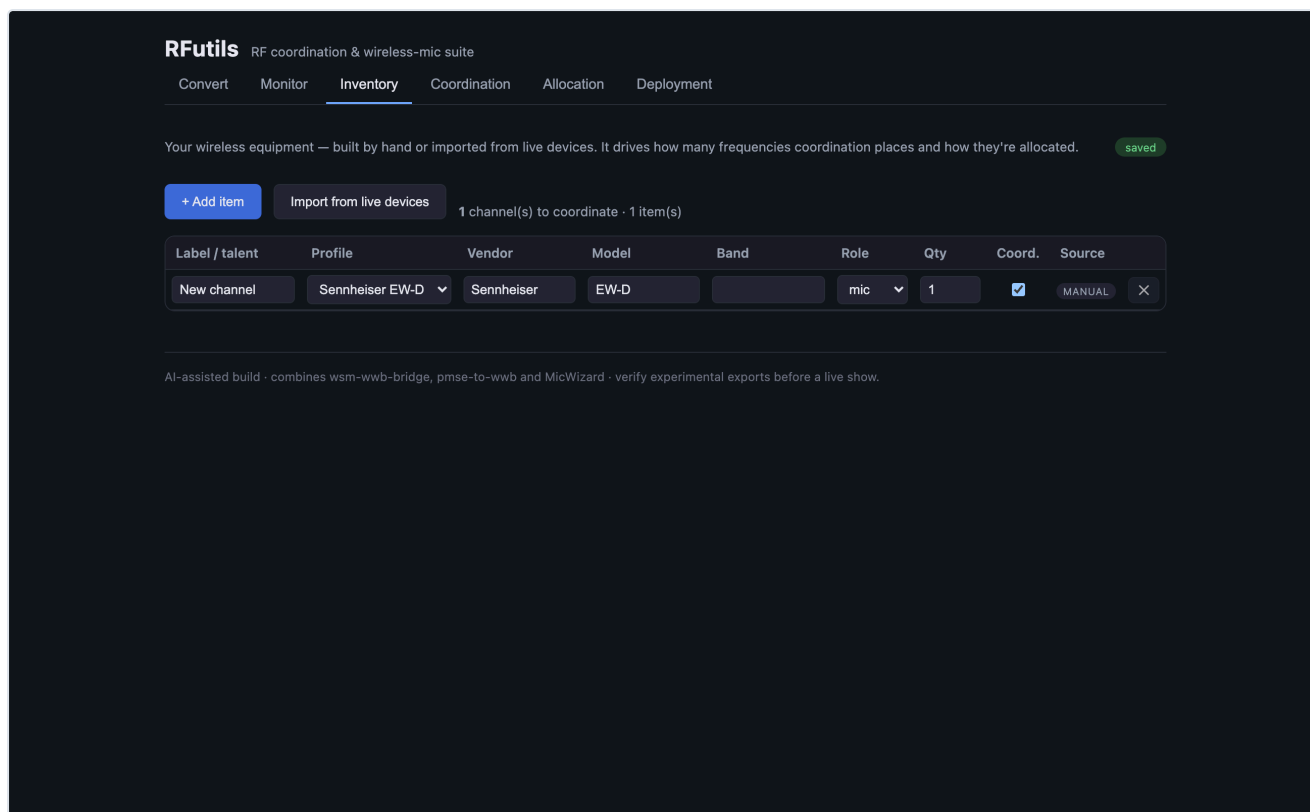
AI-assisted build · combines wsm-wwb-bridge, pmse-to-wwb and MicWizard · verify experimental exports before a live show.

Pick an **equipment profile** and a **band preset** and the spacing defaults follow from the profile — the note under the selector states the occupied bandwidth and recommended spacing it is assuming, and tells you to verify it against your own unit.

Locked / existing is the field that matters on a site where somebody else is already on air: frequencies you put there are treated as immovable and everything else is coordinated around them.

Inventory

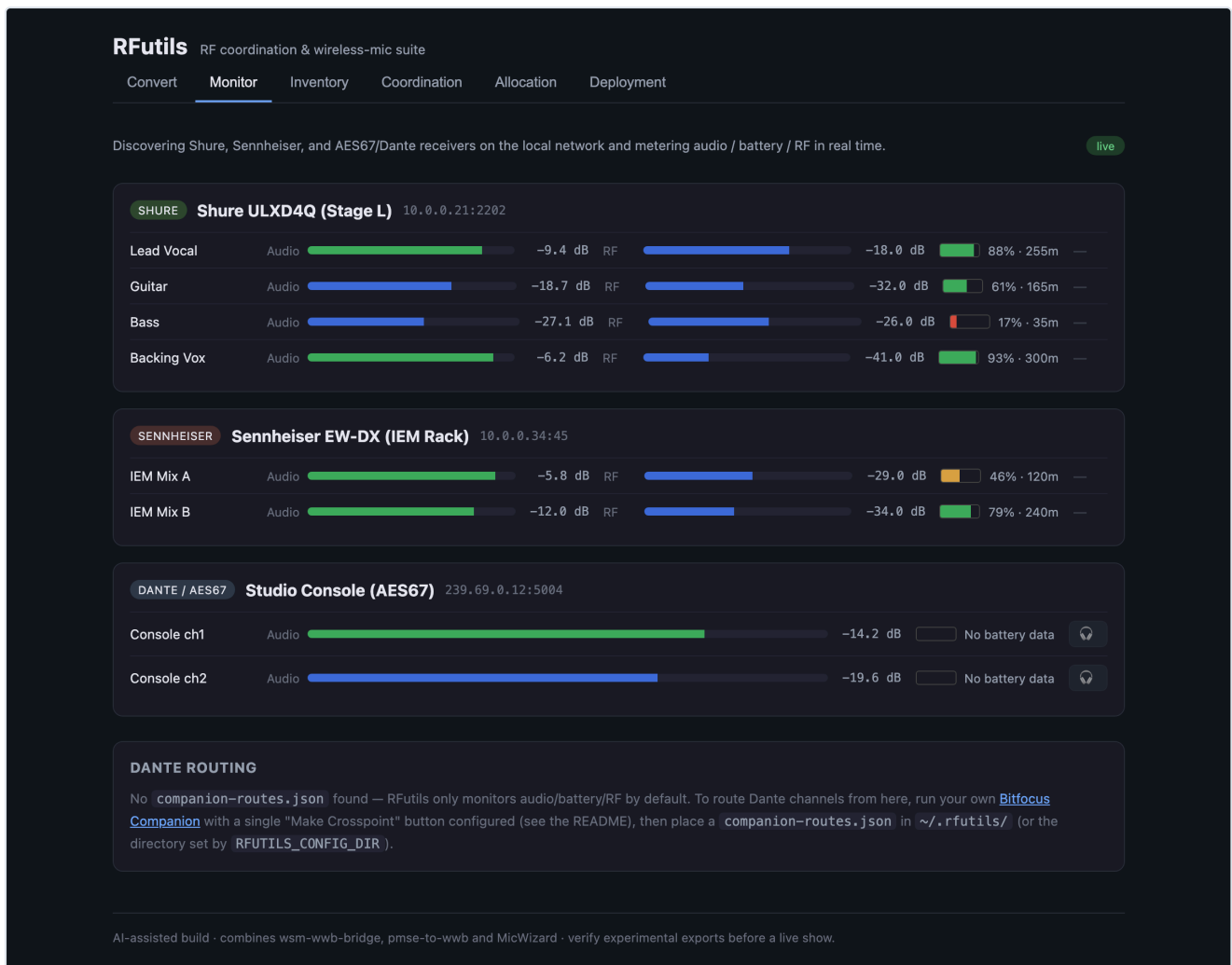
The receivers and transmitters you own, so coordination and allocation have something real to work against.



Saving inventory replaces the whole list rather than merging into it — the app always writes the complete set.

Monitor

Discovers Shure, Sennheiser and AES67/Dante receivers on the local network and meters audio, battery and RF per channel in real time.



Captured against `npm run dev:demo`, so the receivers are simulated.

If nothing appears: RFutils and the receivers must be on the same subnet, discovery must not be blocked by a firewall, and the wireless network must not be isolating clients from each other.

Dante routing is off unless you set it up. RFutils only monitors by default; routing channels from here needs your own Bitfocus Companion with a "Make Crosspoint" button and a `companion-routes.json` in `~/rfutils/`. The panel says so when it is not configured.

Deployment — read this before you press it

Deployment sends commands to real receivers, which may be in use.

RFutils

RF coordination & wireless-mic suite

Convert

Monitor

Inventory

Coordination

Allocation

Deployment

Program allocated frequencies into live receivers, or export files for offline programming.

Export files as

WWB frequency list (.txt)

Download

Refresh devices

#	Name	Frequency (MHz)	Program to device channel
1	Ch 1	606.525	Shure ULXD4Q (Stage L) · Lead Vocal
2	Ch 2	606.925	Shure ULXD4Q (Stage L) · Lead Vocal
3	Ch 3	607.400	— not programmed —
4	Ch 4	608.350	— not programmed —
5	Ch 5	609.050	— not programmed —
6	Ch 6	610.250	— not programmed —
7	Ch 7	610.800	— not programmed —
8	Ch 8	611.825	— not programmed —

Dry-run (2)

Program devices

⚠ Live Shure programming is experimental and untested against hardware — dry-run shows the exact command strings; verify against your receiver's Command Strings PDF before sending.

Channel	Address	Command	Dry run
shure:10.0.0.21:1	10.0.0.21	< SET 1 FREQUENCY 606525 >	would send
shure:10.0.0.21:1	10.0.0.21	< SET 1 FREQUENCY 606925 >	would send

AI-assisted build · combines wsm-wwb-bridge, pmse-to-wwb and MicWizard · verify experimental exports before a live show.

Dry-run shows exactly what would be transmitted — the address, the literal command string, and "would send" against each one — without transmitting anything. **Program devices** is the separate, deliberately distinct button that actually sends.

Live Shure programming is experimental and untested against hardware. The dry-run output exists so you can check the exact command strings against your receiver's own Command Strings PDF before sending. Do that.

Supported control transports today are Shure command strings and the Lectrosonics network protocol. Devices matching neither are listed with no transport and will not be programmed. The same screen also exports files for offline programming, which is the safe route.

Troubleshooting

Symptom	Cause
Nothing is discovered in Monitor	Same subnet, firewall, or client isolation on wireless. Run <code>npm run dev:demo</code> — if simulated devices appear, the app is fine and the problem is the network.
Monitor and Deployment tabs are missing	You are on the hosted build. Those two need the local app.
A CSV imports with the wrong values in the wrong fields	The column mapping guessed wrong. Re-import and set it by hand.
An export doesn't look right in WWB/WSM	Expected occasionally — these formats are reverse-engineered. Please report it with a sample file.
Coordination returns fewer frequencies than asked for	The constraints are unsatisfiable in that range. Widen the tuning ranges, reduce spacing, or drop 5th-order avoidance.
Frequencies clash with someone already on air	Put theirs in Locked / existing and coordinate again.
The port is already in use	Set <code>RFUTILS_SERVER_PORT</code> to something else.
I just want the converters	Set <code>RFUTILS_DISABLE_MONITOR=1</code> , or run <code>npm run dev:verify</code> .

Where your data lives, and who can reach it

Inventory is stored server-side in the local app.

There is no authentication, and the server listens on all interfaces by default. On a shared or untrusted network, bind it to `127.0.0.1` or firewall the port — otherwise anyone who can reach it can read your inventory and **send commands to your receivers**.

See also

- [README](#) — what it is, the three tools it replaces, and downloads
- [wsm-wwb-bridge](#) — the standalone Python predecessor of the conversion tab