

Radio-X12 — Operator Manual

VIOLET AUDIO · BROADCAST CONSOLE · VERSION 2.0

Radio-X12

Operators & Roles

The Console

dMix Connection

Mix-Minus & AUX Routing

Sources & Outputs

RadioBOSS

MIDI Control

Phone-In (SIP)

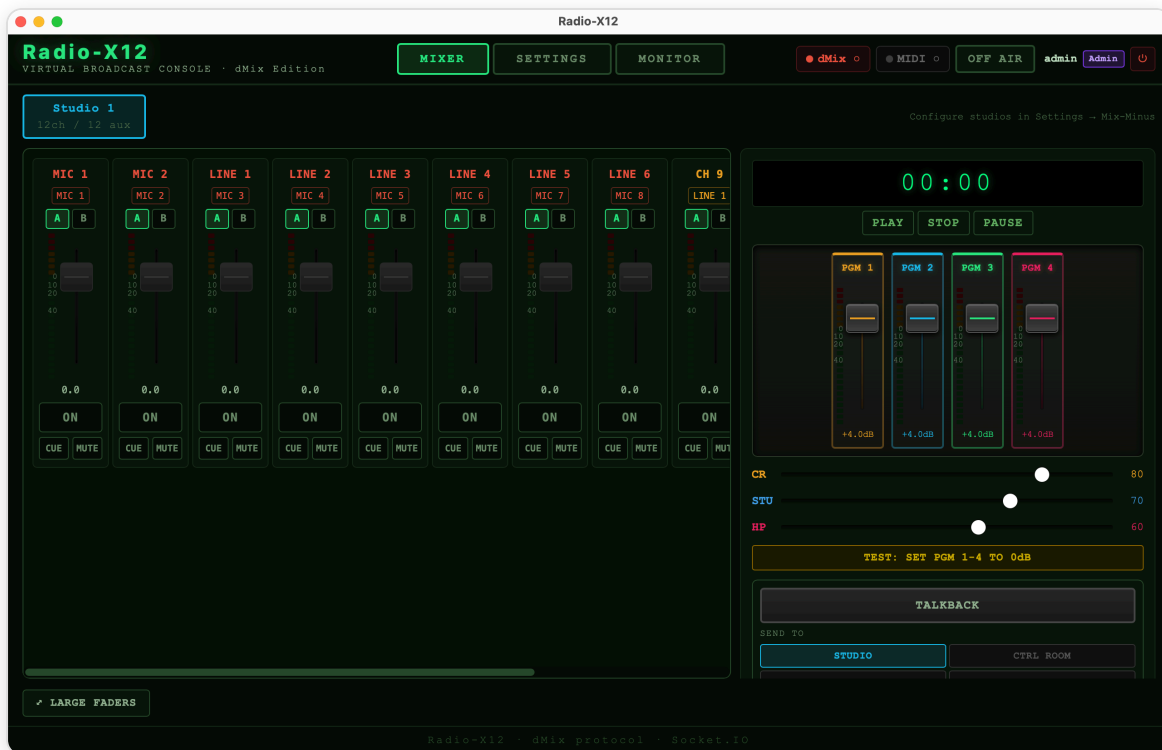
Triggers & Debug

Appearance & Language

Keyboard Shortcuts

Radio-X12

Radio-X12 is a broadcast radio console: a 128-channel mixer surface that controls a **dMix128** audio engine, integrates **RadioBOSS** playout automation, drives relay triggers and MIDI control surfaces, and handles **SIP phone-in** callers — all from one screen. It is a **Violet Audio** product.



The Radio-X12 Mixer page

Starting up

1. **Licence** — on first launch, enter your email to start a free 14-day trial, or enter a **Licence key** and click **Activate**. The trial is tied to this machine, not to local app data.
2. **Sign in** — Radio-X12 then asks for an **operator sign-in**. Each operator has a username, password and a role that decides what they can do. On a brand-new install the first screen instead asks you to **create your Admin account** (username + password); after that you add more operators in Settings → Admin. See Operators & Roles.

The three pages

Across the top of the console are three pages, switched from the centre of the title bar:

- **MIXER** — the channel strips for the selected studio, plus the master section. This is the on-air surface.
- **SETTINGS** — every subsystem, as tabs: Mix-Minus, Sources, Outputs, dMix, MIDI, Triggers, GUI, Debug, Language, RadioBOSS, Phone-In, and (for admins) Admin. Each has its own page in this manual.
- **MONITOR** — an at-a-glance view: program buses, output monitoring, a channel on-air grid, and the broadcast timer.

Which pages and tabs you see depends on your role — a presenter sees only the mixer.

Working offline

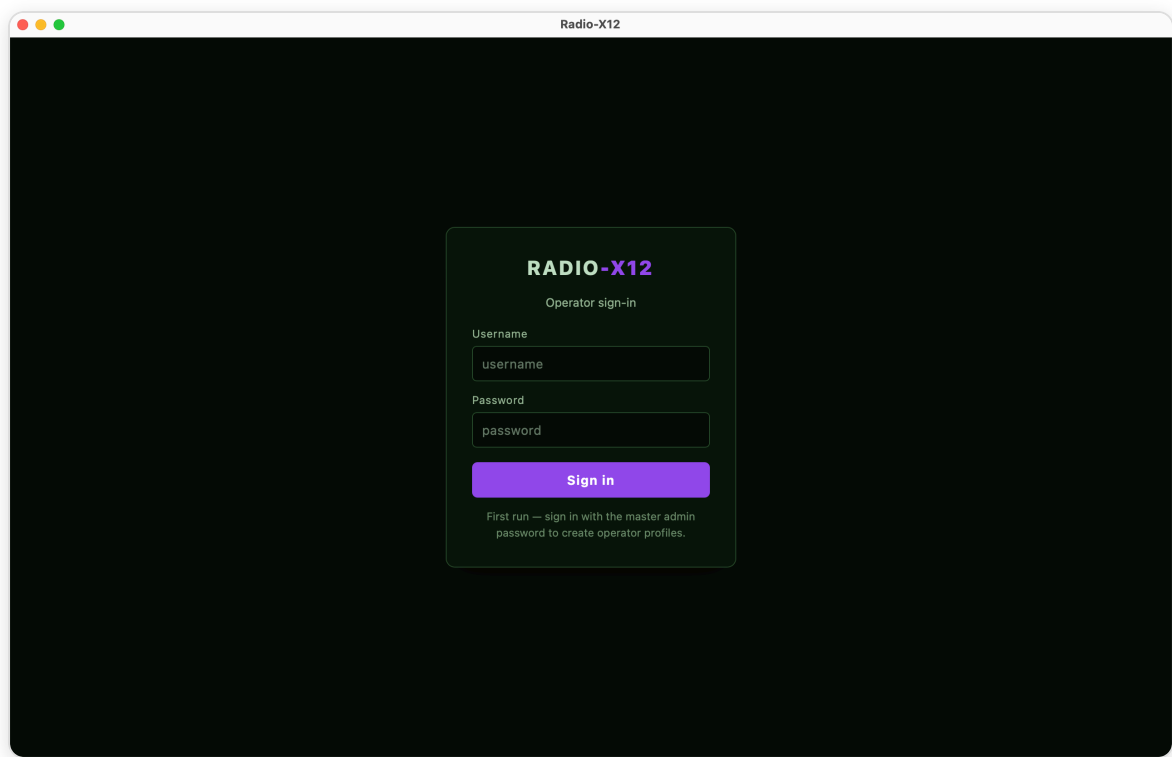
Everything works without hardware connected: fader moves and sends are traced to the **Debug** tab (marked `[no socket]`) instead of going on the wire, so you can lay out and rehearse a show before connecting a dMix engine. Open the **dMix** tab and enter your engine address to go live — see dMix Connection.

Operators & Roles

Radio-X12 signs each user in as an **operator** with a role that controls what they can see and change. This keeps a presenter out of the settings while letting engineers configure the console.

Signing in

After the licence screen, the **Operator sign-in** appears. Enter your username and password. The name and a colour-coded role badge then show in the top-right of the console; the ⏻ button next to it signs out.



Operator sign-in

The three roles

Role	Can access
Presenter	The Mixer page only — faders, ON / CUE / MUTE, sources. No settings.

Role	Can access
Programmer	Mixer, Monitor, and every Settings tab except Admin .
Admin	Everything, including the Admin tab to manage operators.

The console hides whatever a role may not use — a presenter never sees the **SETTINGS** or **MONITOR** buttons, and only an admin sees the **Admin** tab.

First run — create your Admin account

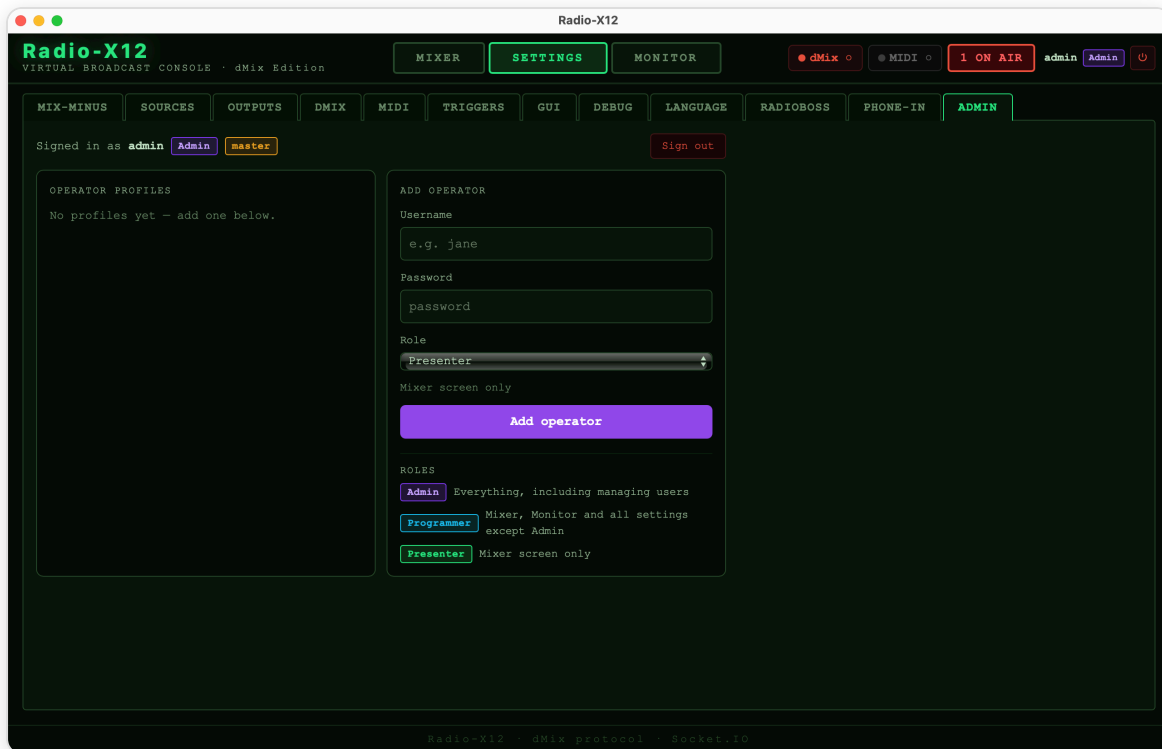
On a fresh install there are no operators yet, so the first screen is **Create your Admin account**: choose an admin **username** and **password** (entered twice to confirm). That becomes the console's first Admin; from there you add the rest of your operators in **Settings** → **Admin**.

The owner also has a **master recovery password** that always grants admin access on any machine, without creating a stored account — a private back door kept for lock-outs, not the normal way in.

Managing operators (Admin tab)

Open **SETTINGS** → **Admin** (visible to admins only):

- **Add operator** — enter a username, password and role.
- **Change a role** — pick a new role from the dropdown next to a user.
- **Reset a password** — click **Password** on a user, type a new one, Save.
- **Delete** — removes that operator.



The Admin panel

What “security” means here

This is a **UI access gate**, not a hardened security boundary. Passwords are stored on this machine as salted SHA-256 hashes (never in plain text), but anyone with developer tools or file access to the machine could get around it. It is the right tool for operator roles on a console; it is not digital-rights protection. If you need enforced security, that belongs on a server, not in the app.

The Console

The title bar carries the station name, the **MIXER / SETTINGS / MONITOR** page switch in the centre, and status on the right: **dMix** and **MIDI** connection chips, an **ON AIR** count, and the signed-in operator.

Mixer page

Channel strips

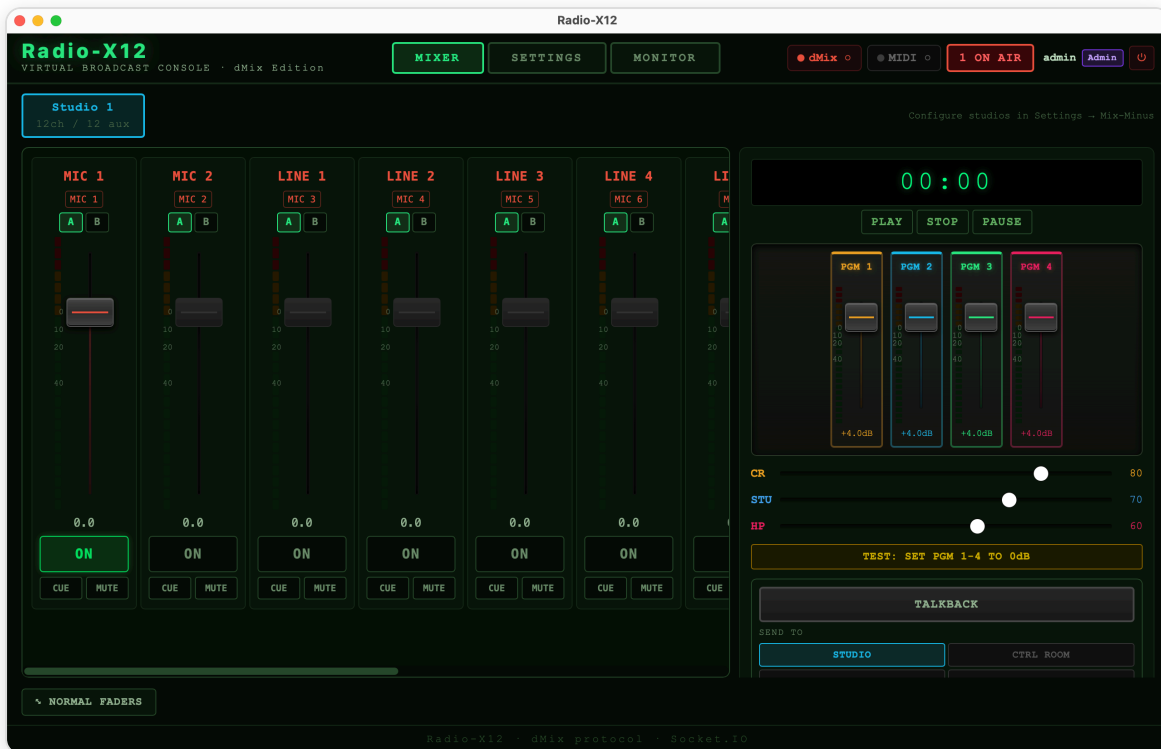
Each strip shows, top to bottom:

- **Label** and the active **source** name, coloured by source group.
- **A / B** source buttons — each channel has two assignable sources; tap to switch which one feeds the fader. Assign them on the Sources page.
- A **VU meter** beside a **fader**. The meter is fed live by dMix once connected and scales to the fader height. Drag the fader; the dB readout below follows the dMix fader law (unity at ~75%).
- **ON** — puts the channel on air (sends `mute = 0` to dMix).
- **CUE** — pre-fade listen (sends `solo`).
- **MUTE** — a local mute state.

Only the **selected studio's** channels appear here. Choose the studio from the bar above the strips; configure studios on the Mix-Minus page. When a studio has more strips than fit, the row **scrolls horizontally**.

Large Faders

The ↗ **Large Faders** button (bottom-left) switches to full-scale, taller and wider faders for touch use. The choice is remembered.



Mixer page with Large Faders

Master section

On the right: the on-air **timer** (PLAY / STOP / PAUSE), four **PGM bus** faders with meters, **CR** / **STU** / **HP** monitor levels, a **hold-to-talk TALKBACK** with destination routing, a **TEST** send, and event / preset buttons.

Settings page

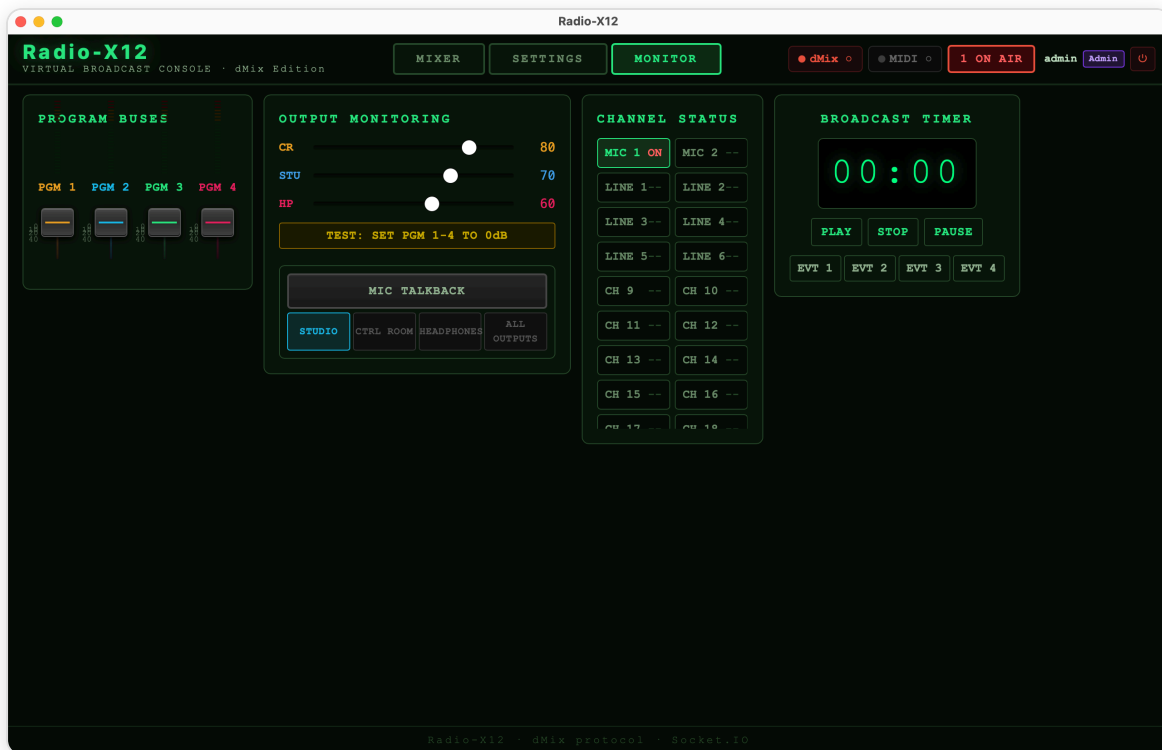
Every subsystem as a tab — Mix-Minus, Sources, Outputs, dMix, MIDI, Triggers, GUI, Debug, Language, RadioBOSS, Phone-In, and (admins only) Admin. Which tabs appear depends on your role — see Operators & Roles.

Monitor page

A wall-style overview:

- **Program buses** — the four PGM faders with live meters.
- **Output monitoring** — CR / STU / HP levels, the TEST send and a compact talkback.
- **Channel status** — an on-air grid of every channel; click one to toggle it on air.

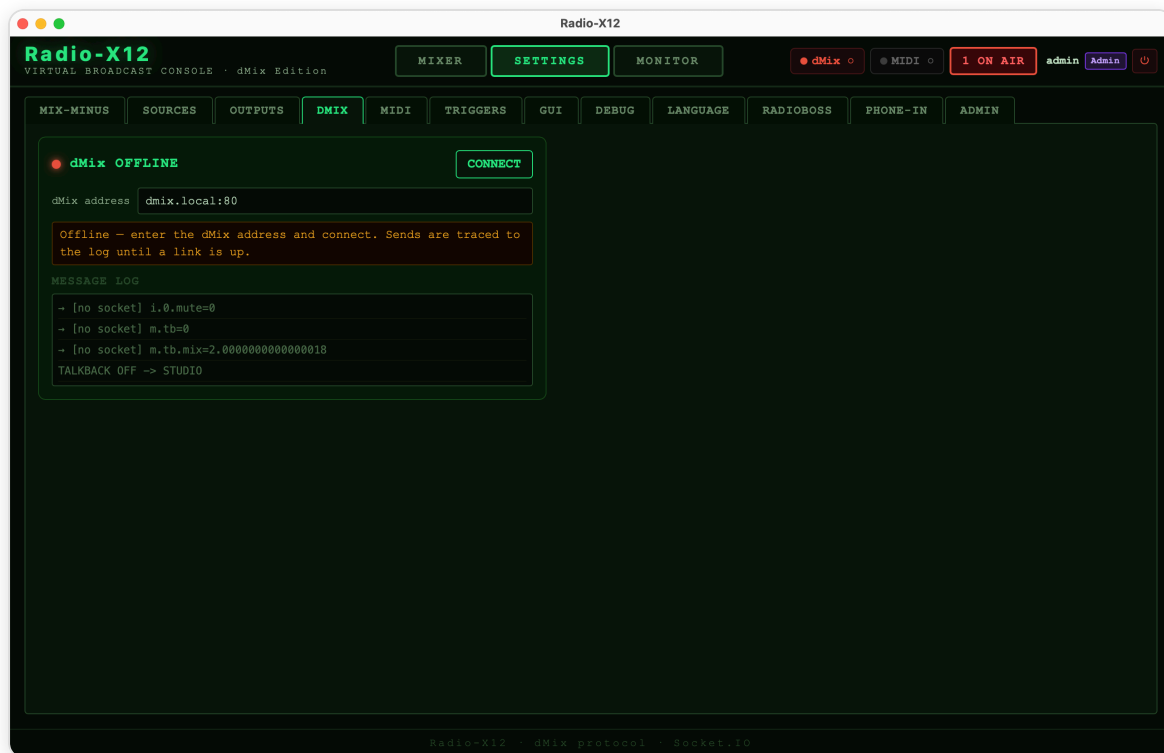
- **Broadcast timer** — a large clock with transport and event buttons.



The Monitor page

dMix Connection

Radio-X12 controls a **dMix128** engine over its native Socket.IO WebSocket — no proxy or bridge software required.



The dMix tab

Connecting

1. Open the **dMix** tab.
2. Enter the engine address — a hostname or IP, optionally with a port, e.g. `dmix.local:80` or `192.168.1.40`.
3. Click **CONNECT**.

The address is remembered and reconnected automatically next launch. If the link drops, Radio-X12 retries every few seconds.

What “connected” means

On connect, Radio-X12 requests a full state snapshot (`INIT`) and applies it to the surface: channel faders, on/off (mute), names, bus assignments, the PGM subgroup levels, output routing and the cue bus all update to match the engine. The header shows **dMix connected** and the dMix tab reads **State synced**.

From then on, moving a fader or toggling **ON / CUE** sends the change to dMix immediately, and the engine's own changes (or another operator's) flow back and update the surface.

V2 firmware (allocation handshake)

dMix **V2** firmware tracks which console currently “owns” the mixer with an *allocation identity*. Radio-X12 handles this automatically: on connect it fetches the current identity from the engine and includes it in the `INIT` , and it waits for the engine to confirm it before it sends any control change. Until that confirmation, outgoing moves are held and the **Debug** log shows them as `[alloc pending]` .

If the engine hands the allocation to another console it sends `ALLOC_INVALIDATED` (or reports `ALLOC_BUSY` during the switch). Radio-X12 pauses writes, discards anything queued, re-fetches the identity and re-synchronises — you'll see these notices in the Debug log. No action is needed; the surface resumes once the engine confirms the new allocation.

Older **V1** engines have no allocation identity, so Radio-X12 connects exactly as before — the handshake simply falls back to the classic `INIT` .

Meters

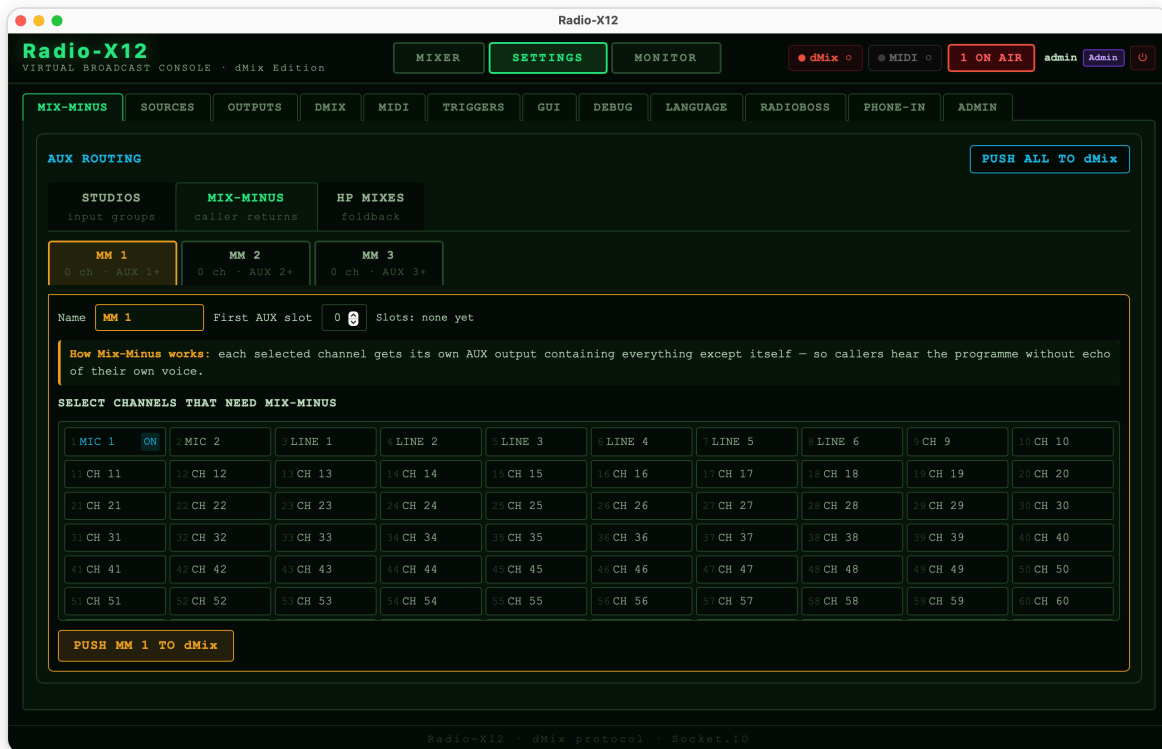
Live **VU** levels arrive over dMix's metering stream and drive every channel meter plus the PGM bus meters, with peak-hold decay.

Offline behaviour

With no link, the surface still works fully for layout and rehearsal — every send is written to the **Debug** log (marked `[no socket]`) instead of the wire. Connecting later brings the surface live with no other changes.

Mix-Minus & AUX Routing

The **Mix-Minus** tab configures AUX sends on dMix. It has three sub-tabs.



The Mix-Minus tab

Studios

A **studio** is a named set of input channels with a block of AUX outputs. Create up to eight, each with its own colour. Pick the input channels that belong to the studio; those become its working set for mix-minus.

Mix-Minus buses

A mix-minus (clean feed) gives a caller *everything except their own voice*, so there's no echo. Select the channels that need a mix-minus return (typically phone, Zoom or IFB returns). Each selected channel is assigned its **own AUX slot** starting at the bus's *First AUX slot*.

The **routing preview** table shows exactly what each AUX carries: `0dB` for every on-air channel, `x` for the owner (excluded), `off` for the rest. Press **PUSH ... TO dMix** to send the

matrix to the engine.

HP mixes

An **HP (headphone / foldback) mix** is a fully custom monitor mix on one AUX bus. Set each channel's send level individually with its slider (or the **off / 0 dB** shortcuts), then **PUSH** it to dMix.

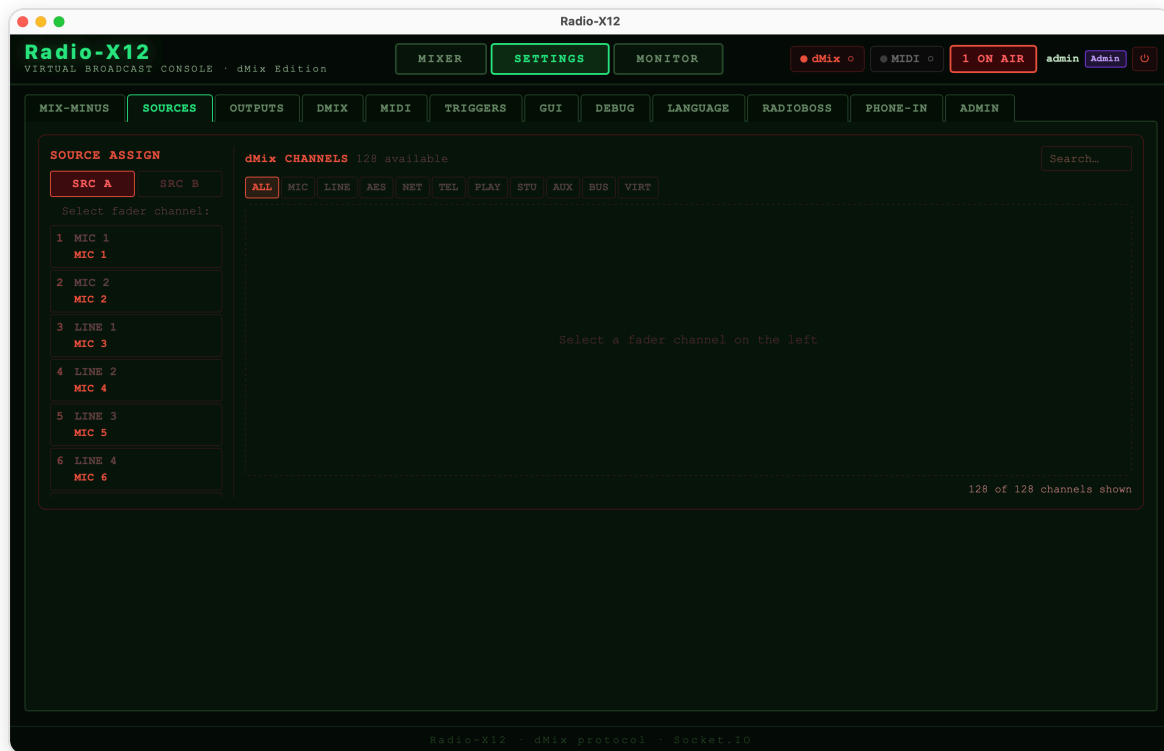
Under the hood

All three write the same dMix keys — `i.<ch>.aux.<bus>.value` (level in dB) and `i.<ch>.aux.<bus>.pre = 0` (post-fader) — so mixes follow the main mix correctly.

Sources & Outputs

Sources

The **Sources** tab maps any of the 128 dMix inputs onto a fader's **A** or **B** slot.



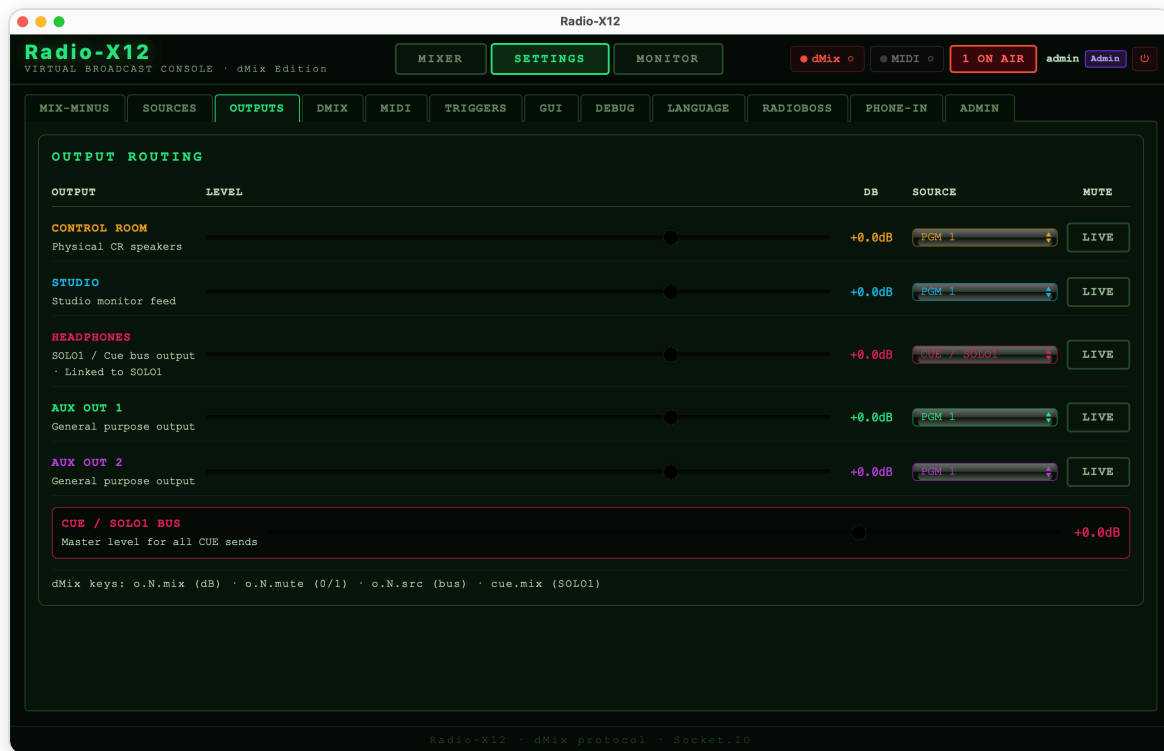
The Sources tab

1. Pick **SRC A** or **SRC B**.
2. Select a fader channel on the left.
3. Click a dMix source on the right. Filter by group (MIC, LINE, AES, NET, TEL, PLAY, STU, AUX, BUS, VIRT) or search by name / number.

Assigning **SRC A** also sets the channel label to the source name. For a **stereo** channel, an **L** / **R** selector appears so you can assign each leg; picking the left leg auto-fills the right with the next dMix input. Tags on each source show where it's already used.

Outputs

The **Outputs** tab is the physical output router: **Control Room**, **Studio**, **Headphones**, and two **AUX** outputs.



The Outputs tab

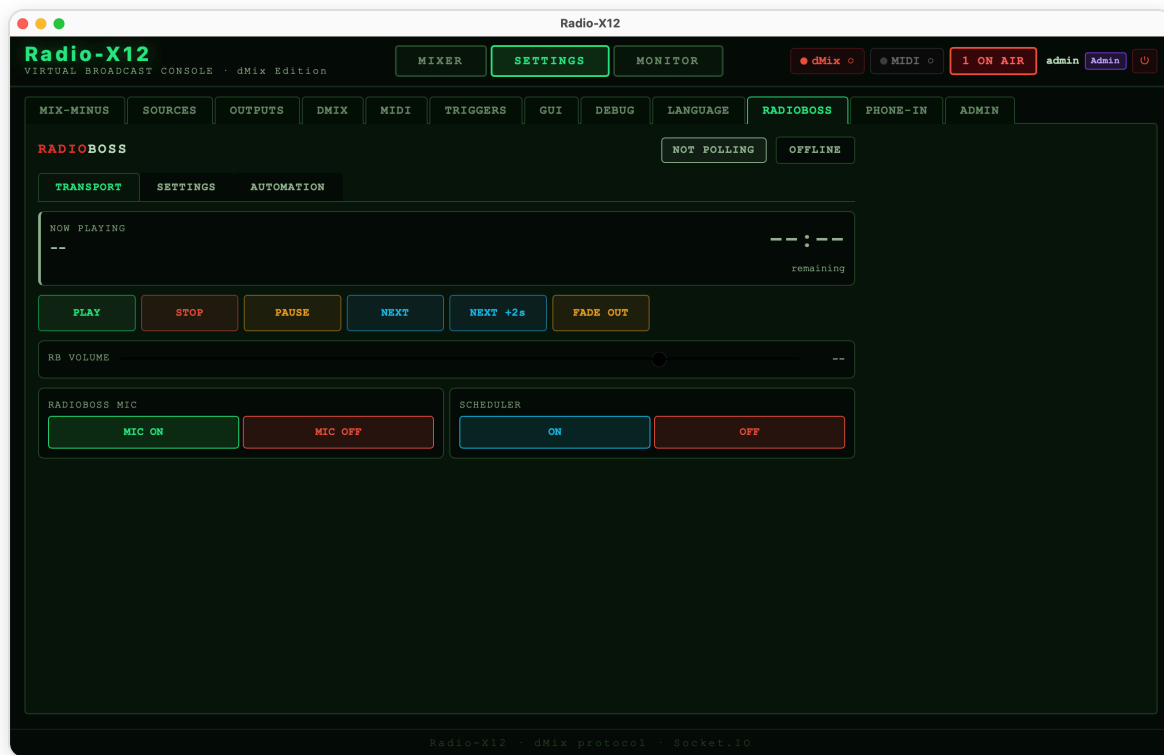
For each output set:

- **Level** — the fader (with dB readout).
- **Source** — which bus feeds it (PGM 1–4, CUE / SOLO1, or Mix-Minus).
- **Mute** — LIVE / MUTED.

Below the outputs, the **CUE / SOLO1 bus** has its own master level for all cue sends. These write the dMix keys `o.N.mix`, `o.N.mute`, `o.N.src` and `cue.mix`.

RadioBOSS

The **RadioBOSS** tab connects to a RadioBOSS playout server's HTTP API. Radio-X12 talks to it **directly** — there is no CORS proxy to run.



The RadioBOSS tab

Settings

On the **Settings** sub-tab, enter the RadioBOSS **IP**, **port** (default 9001), **password**, optional **username**, and a **poll interval**. You can also map two dMix channels: the **RadioBOSS stereo input** and an **announcer mic** channel, used by automation below.

Transport

The **Transport** sub-tab shows live **now-playing** (artist / title), the next track, time remaining and a progress bar. Controls: **PLAY**, **STOP**, **PAUSE**, **NEXT**, **NEXT +2 s**, **FADE OUT**, a **volume** slider, and **mic / scheduler** on-off. Toggle **LIVE** to start polling.

Automation

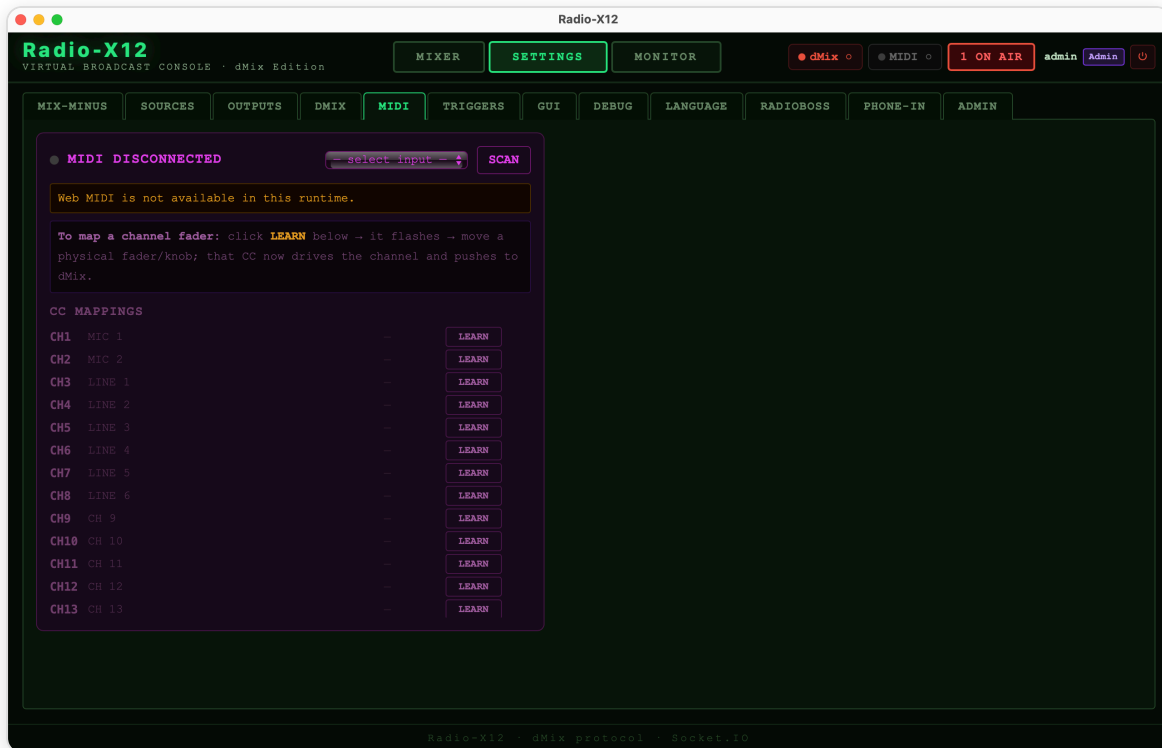
The **Automation** sub-tab fires dMix actions on RadioBOSS state changes (requires LIVE polling):

- Unmute the RadioBOSS channel when playback starts.
- Mute it when playback stops.
- Sync the announcer mic channel to RadioBOSS's mic state.

Custom HTTP triggers let you save arbitrary RadioBOSS API queries (e.g. `cmd=next` or `action=mic&on=1`) and run them on demand.

MIDI Control

Radio-X12 takes hands-on control from a physical MIDI surface. MIDI is handled **natively** by the app (not through the browser), so motorised control surfaces, button boxes and virtual MIDI ports all work — no Web MIDI, no extra drivers.



The MIDI tab

There are two protocols, chosen with the **PROTOCOL** button:

- **MCU** (Mackie Control) — for motorised control surfaces. Faders, buttons, LEDs and scribble-strip displays all work, with two-way feedback.
- **CC LEARN** — generic MIDI-Learn for any fader/knob that sends Control Change messages.

Connecting

Open the **MIDI** tab, click **SCAN**, then pick your controller from the input list. The header shows **MIDI connected**. Your choice is remembered and reconnected on the next launch. Click **DISCONNECT** to release the port.

For motor faders, LEDs and displays to track the desk you also need an **output** — choose it under **FEEDBACK** and set the toggle to **FB ON**.

MCU mode (motorised surfaces)

In **MCU** mode the surface controls eight channels at a time (one *bank*):

- **Faders** drive the eight banked channels and push live to dMix.
- **MUTE** buttons toggle the channel **ON / off**.
- **BANK** ◀ / ▶ page the strips in blocks of eight; the channel buttons nudge one strip at a time. The on-screen strip map shows which channels the surface is currently sitting on.

With a FEEDBACK output set (**FB ON**), the desk talks back to the surface: **motor faders** follow channel levels, **MUTE LEDs** show ON state, and the **scribble-strip LCD** shows channel names.

Radio fader semantics

Radio-X12 maps the surface to broadcast workflow, not studio mixing:

- A station/source fader is **push-to-ON** — bring it up off the bottom and the channel goes **ON**; pull it fully down and it goes **OFF**.
- **SOLO** and **SELECT** are routed to **CUE** (pre-fade listen), and the button LED lights while that channel is cued — so you can audition a source on headphones without putting it to air.
- **MUTE** cuts the channel on the desk itself.

CC LEARN mode (any controller)

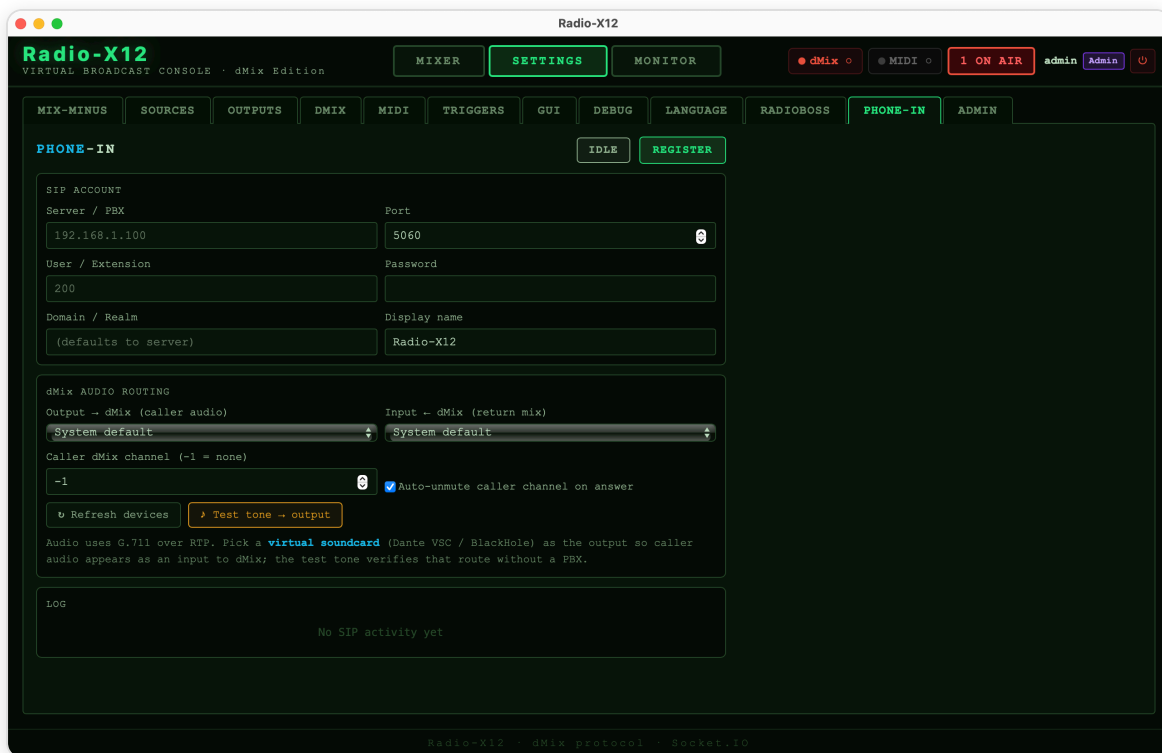
Switch **PROTOCOL** to **CC LEARN** for a controller that sends plain Control Change messages:

1. In the mapping list, click **LEARN** on a channel — it flashes.
2. Move a physical fader or knob.

That CC is now bound to the channel. Moving the control drives the channel fader and pushes to dMix; crossing a small threshold also turns the channel **ON / off**. Set a FEEDBACK output to echo values back on the same CC. Remove a mapping with the **X** button.

Phone-In (SIP)

The **Phone-In** tab is a SIP softphone built into the console for taking callers on air. It registers with a PBX/proxy or SIP trunk and carries audio over RTP using G.711.



The Phone-In tab

Registering

In **SIP Account**, enter your **server / PBX**, **port** (default 5060), **user / extension**, **password**, optional **domain / realm** and a **display name**. Click **REGISTER**. The state badge shows **REGISTERING** → **REGISTERED**.

Making and taking calls

- **Dial**: type an extension or a full `sip:1234@pbx` address and press **CALL**.
- **Incoming**: a call raises an alert with the caller's name — press **ANSWER** or **REJECT**.
- **HANG UP** ends the active call.

Routing audio to dMix

For a caller to be heard on air, their audio has to reach dMix as an input. In **dMix Audio Routing**, choose:

- **Output** → **dMix** — select a **virtual soundcard** (e.g. Dante Virtual Soundcard or BlackHole). Caller audio plays into it, and dMix picks it up as an input.
- **Input** ← **dMix** — the device carrying the studio return mix back to the caller.

Set the **caller's dMix channel** and enable **auto-unmute on answer** to have the channel open automatically for the duration of the call.

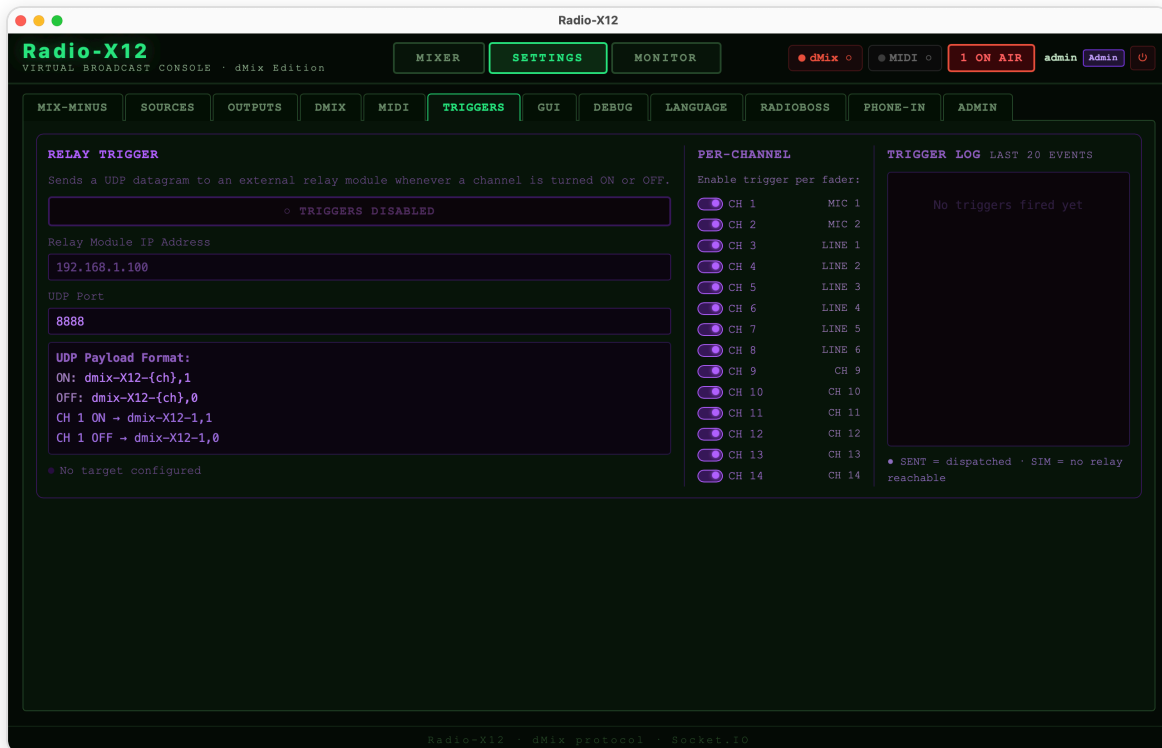
Test tone plays a 440 Hz tone into the chosen output device, so you can confirm the route into dMix without placing a real call.

The first time Radio-X12 opens an input device, macOS asks for **microphone** permission — allow it so the return mix can reach callers.

Triggers & Debug

Relay triggers

The **Triggers** tab sends a UDP datagram to an external relay module whenever a channel goes **ON** or **OFF** — for driving on-air lights, door locks, or other GPIO hardware.

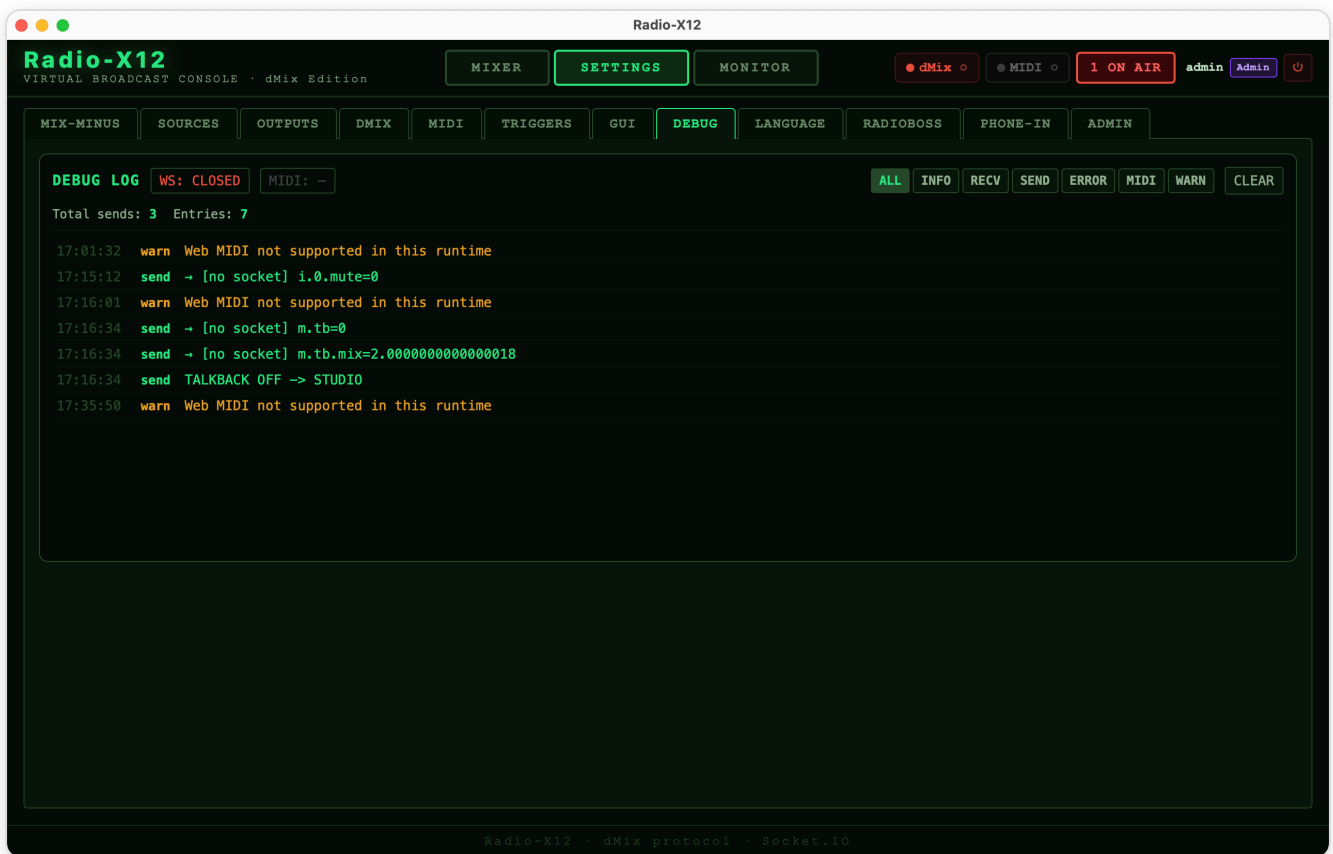


The Triggers tab

Enter the relay module **IP** and **UDP port**, then enable **TRIGGERS ACTIVE**. Each fader can be individually enabled in the **Per-Channel** list. The payload format is `dmix-X12-<ch>,1` (on) and `dmix-X12-<ch>,0` (off). The **Trigger Log** shows the last events, marked **SENT** or **SIM** (no relay reachable).

Debug

The **Debug** tab is a live trace of everything Radio-X12 sends and receives: dMix parameter sends, incoming state, MIDI, RadioBOSS and SIP activity.

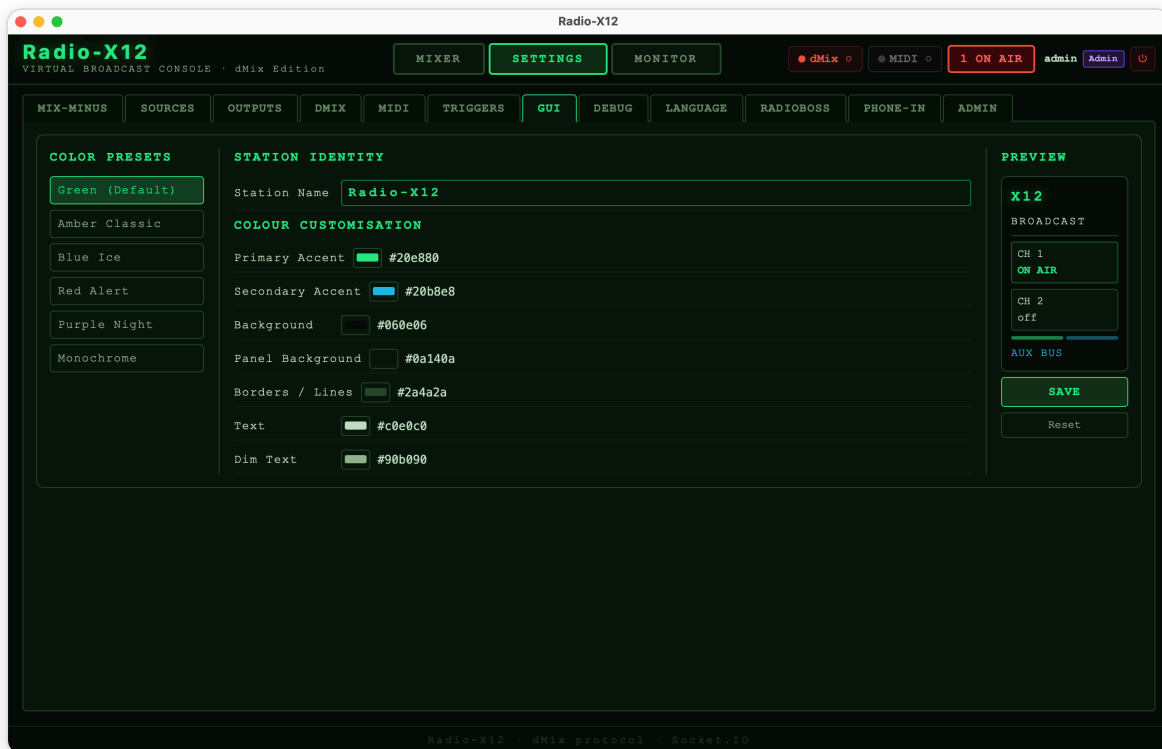


Filter by kind (send / recv / info / warn / error / midi) and **CLEAR** to reset. When something isn't behaving, this is the first place to look — every dMix key written is shown here, including offline [no socket] sends.

Appearance & Language

GUI — theme & station identity

The **GUI** tab customises the console's look:

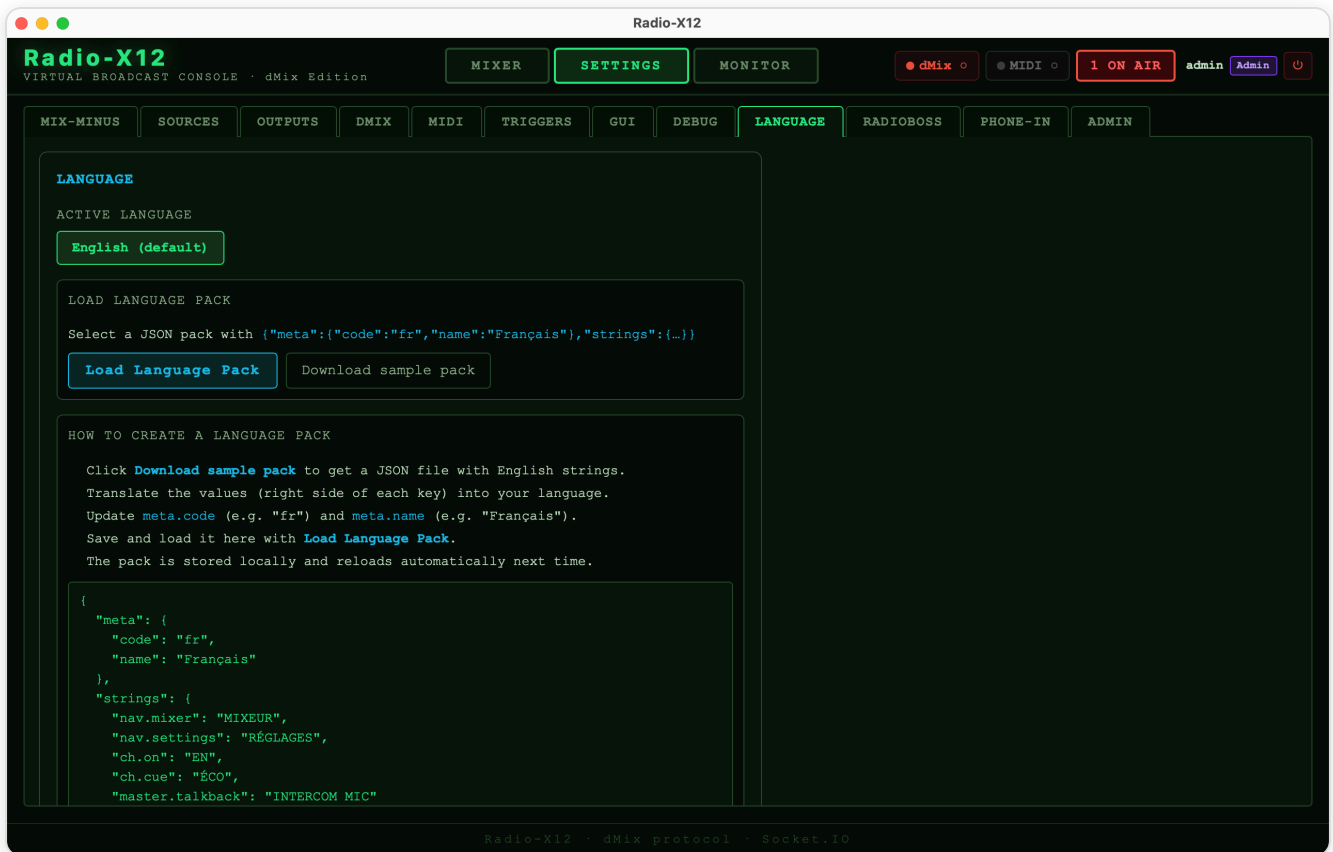


The GUI tab

- **Colour presets** — pick from Green (default), Amber, Blue Ice, Red Alert, Purple Night or Monochrome.
- **Station identity** — set the **station name** shown in the header.
- **Colour customisation** — fine-tune the accent, background, panel, border and text colours, with a live **preview**. **Save** persists your theme; **Reset** returns to the default.

Language

The **Language** tab loads user-supplied translation packs.



Click **Download sample pack** to get a JSON file with the English strings, translate the values, set `meta.code` and `meta.name`, then **Load Language Pack**. Loaded packs are stored locally and re-applied on next launch. English is always available as the default.

Keyboard Shortcuts

Action	Shortcut
Back to the console (from Manual / AI Assistant)	Esc
Settings	Cmd+,
Open this manual	Help menu → Manual

Most of Radio-X12’s controls are on-screen — faders, ON/CUE/MUTE, tabs — and are designed for touch as well as mouse. The native **Help** menu also offers tooltips and a guided walkthrough, and the **View** menu opens the AI Assistant.