



Pilot Tone Trigger

Quick reference — controls & first-run setup

Watches an audio input carrying a continuous pilot tone. While the tone is present the primary source is considered healthy; if the tone drops, the app sends MIDI to the dLive to switch to the backup — an **Action** (Control Change), a **Scene** recall (Program Change), or both. The app can also generate the tone itself (see Tone Generator). Typical tone-drop → MIDI latency with default settings: ≈150–250 ms.

Menu bar icon

Click	Show / hide the control panel.
Right-click	Quit the app (there's no Dock icon or app menu — this is the only way to quit).
Icon colour	The sine-wave mark: grey = stopped. Blue = running, tone present. Red = tone lost — failover sent. Amber = tone restored but latched on backup (press Reset).

Connection

Choose one. Both reach the console the same way underneath — pick whichever needs less new software.

Direct TCP	Opens its own connection to the console. Host must be the MixRack's IP address — never the Surface's — regardless of which unit your network cable is physically plugged into (the Surface bridges through to the MixRack). Default: 192.168.1.70 (the MixRack factory IP); Port defaults to 51325.
Existing MIDI Port	Sends over a MIDI port already routed to the console — most easily the free dLive MIDI Bridge , or A&H's driver / loopMIDI / IAC. Pick it from the dropdown. When the port is dLive MIDI Bridge, this app identifies itself by name ("Pilot Tone Trigger") in the bridge's MIDI Monitor instead of showing as a generic connection.
Base MIDI Channel	Must match Utility → Control → MIDI on the console.
Test Connection	Connects and disconnects once. Green tick = console reachable; red = fix IP/network. Proves reachability, not the console-side mapping — verify that end-to-end once.

Audio Input

Device	The audio interface carrying the pilot tone.
Channel	Which input channel on that device to listen to.

Tone Generator

What it's for	The app can be both ends of the chain: generate the pilot tone to an output, route it through the system under watch (e.g. out to the console and back), and loop it into the Audio Input. If the loop breaks anywhere along the path, the tone disappears and the failover fires.
Output Device	Which audio output carries the generated tone.
Frequency (1000 Hz)	Sine frequency, 20–20 000 Hz. Retunes live, click-free.
Level (–20 dBFS)	Output level. Changes ramp smoothly.
Start / Stop Tone	Independent of the detection engine — the tone keeps running across detection Start/Stop. It never auto-starts on launch; starting it is always a deliberate click.



MIDI Output

What each state change sends. Columns in the app: Signal Lost (red, left) and Signal OK (green, right) — each state has its own settings.

Action (CC)	Sends a Control Change per state — its own CC number and value. On the dLive an incoming CC triggers an ACTION , e.g. setting a group of channels to Preamp A/B/C/D, toggling an insert on/off, and so on. The console must be told what that CC means: map it under Surface → Control → Actions → MIDI Recall , matching the CC number, value and Base MIDI Channel set here. New installs default to CC 85 — Pilot Tone Trigger's own number in the dylanmaudio suite, value 127 for tone present and 0 for tone lost. Existing configs keep whatever number they already had.
Scene (PC)	Sends a Program Change (with Bank Select) per state — recalls a SCENE directly, no mapping needed. Scene numbers 1–500, as shown on the console's Scenes screen.
Both	Fires the Action first, then the Scene, on every state change (and on Reset).

Important — a Scene recall changes everything the Scene stores, not just what you intend. Set each Scene's Recall Filter to allow only the items the failover should touch. For Actions, nothing happens at all until the matching MIDI Recall mapping exists on the console.

Failback

Automatic	When the tone returns, the Signal OK output fires by itself and the system is back on primary. Switching from Latch to Automatic releases an armed latch — and fires the Signal OK output immediately if the tone is already back.
Latch until Reset	After a dropout the app stays on backup even when the tone returns — no automatic flap back onto a source that just failed. While latched, further drop/restore cycles send nothing at all (the console is already on backup) until Reset or a switch to Automatic releases it. Recommended for shows.
Reset button	Lights up (amber) once the tone is back while latched. Sends the Signal OK output and re-arms. Refuses if the tone is still below threshold.

Level meter

Amber line	Threshold — the level must be above this to count as tone present (recover point).
Red line	Trip point (Threshold – Hysteresis) — the level must fall below this to count as tone lost.
Between the lines	State holds; the status caption reads “In hysteresis band” and shows the exact trip level.

Sensitivity

Defaults are tuned for a continuous pilot tone.

Threshold (–45 dB)	Set below the tone level. Raising it closer to the tone speeds up detection.
Dropout Hold (50 ms)	Time below the trip point required before failing over (rides out single glitched audio blocks).
Hysteresis (6 dB)	Gap between recover and trip lines — prevents flapping at the threshold.
Min Retrigger (200 ms)	Minimum time between consecutive switches, even if the level keeps crossing.
Env Release (30 ms)	How fast the level reading falls — the main latency lever. Shorter = faster detection.

Important — watching programme material instead of a tone? Raise Dropout Hold above the material's longest natural silence and lengthen Env Release (~300 ms), or quiet passages will trip a failover.



Running it

Start / Stop	Opens (or closes) the audio input and begins (or stops) watching the level. The app arms assuming the tone is present — a feed that is already dead trips the failover after Dropout Hold.
Status tile	Stopped / Signal Present / Signal Lost / Restored — Latched. Mirrors the menu bar icon colour. A red “Start Failed” tile names the failing subsystem (console / audio input) in full.
First launch	macOS will prompt for microphone access the first time you click Start — allow it, or the app won't receive any audio.
Settings	Persist to <code>~/Library/Application Support/Pilot Tone Trigger/config.json</code> .

Bitfocus Companion

Loopback only, port 8772, no setup needed on this side — add this app in the dylanmaudio Companion module.

What it controls	Run, Failback mode, Reset, and the tone generator — as Companion buttons, with this app's own state shown back on them (works from Companion even if the popover isn't open).
Allow Companion control	On by default. Off refuses every Companion press; Companion can still show why.
Lock show-critical controls	Off by default. On refuses Run-off and Failback mode from Companion — Reset and the tone generator are never locked.

If something goes wrong

Export log	Click Export log in the panel footer. A copy of this session's log lands on your Desktop as a text file — send that with your report. Every launch is logged automatically to <code>~/Library/Logs/dylanmaudio/Pilot Tone Trigger/</code> ; nothing to switch on.
Send feedback	Also in the footer — a short message straight to me, no email client needed.

What's new in v1.0.5

- Tidier popover: the Bitfocus Companion settings and the on-screen hints now sit behind disclosures, closed by default (a green dot on the Companion header shows when Companion is connected); the trademark notice at the bottom has gone — it's in the Quick Reference and the licence. Companion can now also bring the app's window forward.
- New: this app can now be controlled from Bitfocus Companion — Run, Failback mode, Reset, and the tone generator all show up as Companion buttons, with the app's own state reflected back on them. On by default; two new switches in the app let you turn Companion control off or lock the show-critical ones.