

ORRA AUDIO · PRODUCT MANUAL

Version 1.0.0 (beta)

ORRA VISUALIZER

AUDIO-REACTIVE VISUAL ENGINE

Layered, audio-reactive visuals — for producers, performers, and music lovers.

Your sound, visualized.

Welcome

Orra Visualizer is a layered, audio-reactive visual engine for macOS. Point it at your sound — a DAW, Spotify, a microphone, anything your Mac can hear — and compose live visuals from webcams, screen captures, video, text, and an open ecosystem of GPU shader plugins, all reacting to your audio in real time. Record it, stream it, or run it full-screen behind your set.

It's built for music producers, performers, and streamers — but you don't have to be one. If you love music, it turns listening into something you can watch.

Making your own visuals is the heart of it. The plugin system is an open framework, and its built-in templates teach your favorite AI everything the engine can do — so creating a plugin is a conversation: describe the visual you want, paste the result into the built-in editor, and watch it run live on your canvas. No graphics programming required. Most of what people run in Orra Visualizer, they made — or someone made and shared as a file (§15 covers creating, §13 covers sharing).

This manual covers everything the app can do. Skim the table of contents, or read straight through — it's organized in the order you'll meet things.

Contents

Getting started

The interface

Projects

Scenes

Layers & sources

Working the canvas

Managing layers

Layer properties

The mixer & audio sources

Orra Link

Transport control

Audio reactivity

Plugins

The plugin editor

Creating your own plugins

MIDI control

Global FX

Recording

Streaming

Display & performance

Keyboard shortcuts — complete reference

macOS permissions

Licensing

Your files & backups

Troubleshooting

1. Getting started

First launch

1. **License gate.** The first thing you'll see. Three options: **Enter License Key**, **Start 3-Day Demo** (one demo per machine), or **Purchase** (opens orraaudio.com). Activate or start the demo to continue; closing the gate quits the app.
2. **The home page.** Orra Visualizer opens to the **Project Browser** — a tiled grid of your projects. On a fresh install it says "Welcome to Orra Visualizer." with **+ New Project** and **Open File...** buttons.
3. **Content installs separately.** Orra Visualizer ships as an engine, not a library — nothing is baked into the app. Plugins and scenes arrive as **packs** (`.orrviz-pack`) or as individual `.js` files that you drop onto the window; see §13. Until you install some, the Generative and Particles menus are empty and the only scene transitions available are **None (hard cut)** and **Default (fade)**. Everything you install is yours to edit or delete, and nothing is ever overwritten behind your back. Orra Audio publishes packs and plugins to download — grab one, drop it on the window, done — or skip straight to creating your own (§15).
4. **High-resolution displays.** If you're on a 4K/5K screen, a one-time prompt offers **Auto render scaling** — say yes if you're not sure; it keeps frame rates smooth by adapting internal resolution (your recordings are unaffected). See §20.
5. **Permissions arrive when needed, never up front.** The first webcam layer asks for Camera; the first system-audio source asks for Screen Recording; the first Spotify transport click asks for app control. If you ever deny one, **Help** → **Permissions** is the rescue path. See §22.

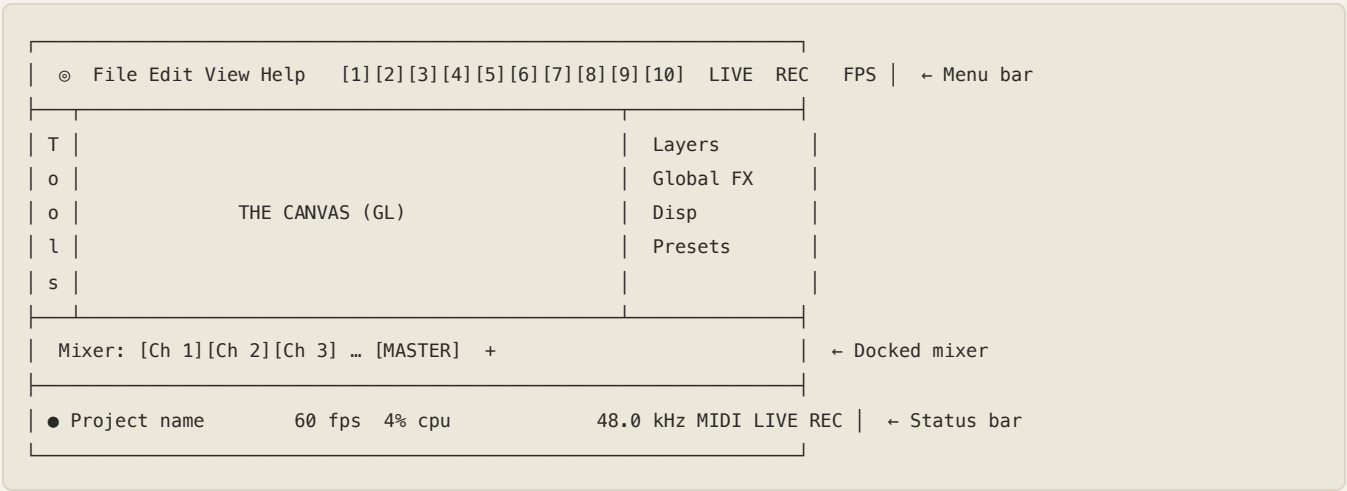
System requirements

- **macOS 13 (Ventura) or later.**
- **Built for Apple Silicon.** Intel Macs run it — the app even auto-tunes itself for them — but performance and behavior may vary, and Intel support is best-effort during the beta.

Your first two minutes

1. **+ New Project** → name it, pick a resolution (1920×1080 default), and optionally pick an audio source right in the dialog.
2. Add audio: open the mixer (**Cmd+M**), click **+**, and pick **System Audio** → **Application** → **Spotify** (or your DAW, or an input device).
3. Add a visual: right-click the canvas → **Add Generative** → pick anything in the list. (Nothing there yet? Install a pack or drop a `.js` plugin onto the window first — §13.)
4. Play some music. It moves. Welcome to Orra Visualizer.

2. The interface



- **Menu bar** — File/Edit/View/Help menus, the ten **scene squares** (click to switch, right-click for scene options), the **LIVE** and **REC** buttons, and an FPS readout (shows `75% · 60` style when render scaling is active).
- **Toolbar** (left) — two zones: **MODE** (Select, Crop, Lock) and **ADD** (Webcam, Screen Capture, File, Text, and a combined Generative `{ }` button covering generative *and* particle sources). The buttons are icon-only; hover for a tooltip naming each one.
- **The canvas** — your composition. Everything here is the output.
- **Right sidebar** — four tabs: **Layers** (the stack), **Global FX** (scene-wide effects), **Disp** (display settings), **Presets** (browser).
- **Docked mixer** (bottom) — audio channel strips. Toggle with **Cmd+M**; drag its top edge to resize.
- **Status bar** — project name (gold dot = unsaved changes), FPS/CPU, sample rate, and MIDI/LIVE/REC activity dots.

Useful global switches:

Action	How
Hide/show the side panels	S or Tab , or View → Toggle Panels
Hide/show the mixer	Cmd+M , or View → Toggle Mixer Panel
Performance View — canvas only, full screen	Ctrl+Cmd+F (View → Full Screen Canvas). All chrome collapses; exit restores your exact layout. Mouse and shortcuts still work — it's the full app minus chrome, perfect for projection or performance.
Keep Orra Visualizer above other apps	View → Always On Top (also in the Disp tab)
Hover tooltips on/off	View → Tooltips → Show Tooltips (remembered across launches)
Tooltip language	View → Tooltips → System Default / English / 日本語 / Español / Português / Nederlands . Takes effect on the very next hover — no relaunch.
Change the app's colours	View → Theme, or the Disp tab's Appearance section (\$20)

About the languages: the tooltip text itself is translated into Japanese, Spanish, Portuguese, and Dutch. Coverage is partial and growing — anything not yet translated falls back to English rather than going blank. Menus, labels, and buttons are English-only for now; this setting covers tooltips.

3. Projects

A project is a single `.ovz` file (macOS shows it as one icon). Inside it: ten scenes, your mixer setup, output resolution, and a thumbnail. Projects save by default to `~/Library/Application Support/OrrViz/Projects/`.

Action	Shortcut	Notes
New project	Cmd+N	Opens the setup dialog: name, aspect/resolution (presets, Free Resize, or Custom), and an optional audio source.
Open	Cmd+O	Also File → Open Recent (last 10).
Save	Cmd+S	Silent re-save. Falls through to Save As for an unsaved "Untitled".
Save As	Cmd+Shift+S	
Close project	Cmd+W	Returns to the home page — never quits the app. (The red traffic-light button quits.)
Browse projects	Cmd+Shift+O	The home page — see below.
Capture thumbnail	Cmd+Shift+T	Grabs the current canvas as the project tile ("hero shot"). Once you capture one manually, auto-saves stop replacing it.

Unsaved changes. The window title shows `*` and the status bar a gold dot when the project is dirty. Quitting or closing with changes prompts **Save / Don't Save / Cancel**. Undoing back to the last-saved state clears the dirty flag on its own.

Crash recovery. While a project has unsaved changes, Orra Visualizer continuously writes a recovery snapshot. If the app didn't shut down cleanly, the next launch offers **Restore** (brings back your unsaved work, reconnects the project identity) or **Discard**.

The project browser

The home page lists your projects two ways — flip between them with the **view toggle**:

- **Thumbnail mode** — a grid of project tiles. The **thumbnail size slider** next to the toggle scales the tiles, and **Cmd+scroll** over the grid does the same thing without leaving the mouse. (Plain scrolling just scrolls the list.)
- **List mode** — fixed-height rows. The size slider hides itself here, since there's nothing to resize.

Across both: a live **search** field, a **Sort** menu (**Modified / Name / Created**), and ★ **favorites** pinned to the top of the grid under their own header.

Right-click any project for **Open**, **Reveal in Finder**, and **Move to Trash**. Trashing asks first, with a **Move to Trash & Don't Ask Again** option, and refuses outright if that project is the one you currently have open.

Renaming and duplicating projects happens in Finder — a `.ovz` is a normal file you can rename or copy. The browser reflects it.

4. Scenes

Every project holds **ten scenes** — full snapshots of your layer stack, global FX, grid settings, colors, and MIDI mappings. The **mixer, output resolution, and Optimize Performance are project-wide** (shared by all scenes).

- **Switch scenes:** keys **1–9** and **0** (scene 10), or click the squares in the menu bar (the switch happens on *mouse-up*, so a drag never triggers one). Projects always open on scene 1.
- **Swap two scenes:** **drag** one square onto another — the two slots trade places. Right-click → **Move Scene to** → **Scene N** does the same swap.
- **Copy a scene:** **Option+drag** one square onto another, or right-click a square → **Copy Scene to** → **Scene N**. (Option+*click* without dragging does nothing — it won't switch scenes.)
- **Clear a scene:** right-click its square → **Clear Scene**. It asks first, and the dialog offers **Clear**, **Clear & Don't Ask Again**, or **Cancel**.
- **Share a scene:** right-click → **Share Scene** exports the single scene as an `.orrviz-pack`, carrying the plugins it uses. **File** → **Export Scene** is the same exporter. See §13 for what happens when someone drops one back in.
- **Scene presets:** **File** → **Save Scene as Preset** captures the whole scene (with thumbnail) for reuse in any project; load from **File** → **Load Preset** or the **Presets** tab, which also supports drag-and-drop onto the canvas.

Transitions

Right-click a scene's square → **Transition**. The transition plays when you switch *away* from that scene.

- **Open Plugin** sits at the top of the submenu — it opens the transition's own editor. It's greyed out (and says so) until you've picked a transition plugin. **Cmd+click the scene square** is the shortcut for the same thing; on a scene set to None or Default it simply does nothing.
- Below that: **None (hard cut)**, **Default (fade)**, then every transition plugin you've installed, a **Duration** submenu (**0.3 / 0.6 / 1.0 / 1.5 / 2.0 s**, default 0.6), and **Create Transition** to write your own.

Transitions have real, editable parameters. The Open Plugin window is an ordinary plugin editor — sliders, custom UI, presets, its own Cmd+Z undo — and it behaves the way you'd want while you're dialling one in:

- Edits apply **mid-blend**. Leave the window open, tap between two scenes, and drag a slider while the blend is running.
- **Each scene keeps its own values.** Scene 3's Glass Shatter can be tuned completely differently from scene 7's.
- **Picking a different transition resets** that scene's parameters (changing only the duration keeps them).
- The window **stays open across scene switches** — that's the test loop. It does close if you reshuffle the slots underneath it (Clear Scene, a swap, a copy, or loading another project).

MIDI Learn is not available in the transition editor — transition parameters sit outside the layer-keyed mapping model, so the menu is hidden rather than offering a mapping that wouldn't work.

5. Layers & sources

Everything on the canvas is a **layer**. Add them three ways: the toolbar **ADD** zone, the sidebar **+ Add Layer** menu, or **right-clicking empty canvas**.

Right-clicking the canvas leads with the creative sources — **Add Generative** and **Add Particles** sit at the top, above Webcam, Screen Capture, File, and Text. Both of those submenus end with a **Create Generative** / **Create Particles** item that drops you straight into the Plugin Creator (§15), and both offer **New Folder** for organising as you go. The sidebar's **+ Add Layer** menu covers the same ground with slightly shorter labels — **Sources** (Webcam / Screen Capture / File / Text), **Generative**, **Particles**, **Composite FX** — and keeps its own **Create Particles** / **Create Generative** items.

Source	What it is	Good to know
Webcam	A live camera layer	The add-menu lists every camera by name; pick your device. Per-layer camera choice. Supports background removal, hand tracking , and face tracking — see §8.
Screen Capture	A live capture layer	Three kinds: a whole display , one specific window , or an application (its largest window). An app with no open window can't be captured. Retarget it later from the layer's Source dropdown without re-adding it.
Video / Image file	Media from disk	PNG, JPG, GIF, BMP, TIFF; MP4, MOV, M4V, AVI. Or just drag a file onto the canvas — it lands where you drop it (drag-drop accepts PNG/JPG/MP4/MOV/M4V).
Text	Vector text you type on the canvas	Select the Text tool, then drag on the canvas — the drag height sets the size (12–200 px). A quick click places default size. Type immediately; Enter commits, Esc cancels.
Generative	A shader plugin that <i>generates</i> visuals	The heart of Orra Visualizer. Pick from the Generative menu — everything you've installed or written, grouped by folder. Includes 3D mesh plugins.
Particles	GPU particle-system plugins (up to 1M particles)	Their own submenu. Audio-reactive by design.
Composite FX	A layer that processes everything <i>below</i> it	Put one on top and give it an FX chain — it treats the whole mix beneath as its input.

Layers stack bottom-to-top like any compositor. Meters (spectrum, levels) live in the **mixer**, not as layers.

6. Working the canvas

Select mode (the pointer — default)

- **Move**: drag a layer. **Resize**: drag a corner handle. **Rotate**: **Cmd+drag** a corner handle.
- **Click-through stacks**: clicking the same spot repeatedly cycles selection downward through overlapping layers.
- **Multi-select**: **Shift+click** or **Cmd+click** toggles layers into the selection; drag a **marquee** on empty canvas to select a region.
- **Nudge**: arrow keys move selected layers 1 px; **Shift+arrows** 10 px.
- **Center a layer instantly**: hold **Shift+C** and click it.
- **Scale from the wheel**: hold **Option** and scroll over a selected layer.
- **Lock**: the toolbar Lock button, the padlock on a layer row, or right-click → Lock. Locked layers ignore move/resize/rotate/nudge.

Grid, snap & safe zones

- **G** toggles the grid; **Shift+G** toggles snapping.
- The **Disp** tab holds Grid/Snap toggles plus **Density** (2–100) and **Ghost** opacity sliders, and a **Safe Zones** toggle (broadcast-safe guides).

The crop tool

Click **Crop** in the toolbar and choose a mode:

- **Rectangular** — drag any *corner handle* of the selected layer to trim it. PowerPoint-style: the layer stays fully live; move/resize afterwards keeps the crop.
- **Lasso (pen)** — click the layer to drop the first anchor, click to add points; a gold rubber-band follows. Close it by clicking the start point, **double-clicking**, or pressing **Enter**. **Esc** cancels, **Backspace** removes the last point, **Alt+drag** draws a freehand run of points for curves.
- **Freehand** — click-drag to draw the outline in one stroke; releasing commits.

Crop works on **every** source type. Undo it any time with right-click → **Reset Crop**.

7. Managing layers

Each row in the **Layers** tab shows a type badge, the name, and a cluster of icon buttons:

Icon	What it does
Pencil — <i>Open Plugin</i>	Opens the layer's plugin editor (or the text editor on a text layer). Only appears on layers that have something to edit: generative, particles, 3D, canvas, and text.
Headphones	The layer's audio-reactivity source (§12). Always present on every layer, and its tooltip reads out the current routing and whether it's live — e.g. "Master - Live".
Padlock	Locks the layer: no move, resize, or rotate on the canvas.
S	Solo — hide every other layer. Right-click to MIDI learn.
M	Mute — hide this layer. Right-click to MIDI learn.
Action	How
Rename	Right-click → Rename. Renaming a text layer changes its on-screen text.
Reorder	Drag rows up/down (multi-selections move as a block), or Cmd+[/ Cmd+] .
Select all / none	Cmd+A / Cmd+Shift+A (also Edit → Select All / Deselect All). Clicking dead space on the toolbar, menu bar, status bar, or mixer also clears the selection.
Duplicate	Cmd+D . The copy carries nearly everything — transform, opacity, blend, keying and background removal, crop, FX chain, audio routing — and it carries the layer's MIDI bindings too , so one physical knob drives both copies in lockstep until you re-learn on one of them. Duplicates always start unlocked .
Copy / Paste layer	Cmd+C / Cmd+V
Delete	Delete , or right-click → Delete Layer
Save as preset	Right-click → Save Layer as Preset (reusable via File → Add Layer from Preset)

Right-click a layer on the canvas for the full menu: add effects, flip horizontal/vertical, reset rotation, **Fit to Canvas**, reset transform/crop, bring to front / send to back, lock, **Copy Properties / Paste Properties** (carries opacity, blend, render flags, background-removal settings, and the entire FX chain to other layers), **Copy/Move to Scene** (any of the 10), save as preset, and delete.

8. Layer properties

Select a layer and the properties panel (below the layer list) shows collapsible sections. What appears depends on the source type:

- **Source** — a device-aware picker that changes with the layer type:
 - On a **webcam** layer it lists your Mac's cameras under a **Camera** heading, with the one currently in use ticked.
 - On a **screen-capture** layer it lists **Displays**, **Windows**, and **Applications** (empty groups are omitted), and picking a different one **retargets the layer live** — no need to delete and re-add.
 - On **Image** and **Text** layers the Source row is **hidden entirely**, since there's nothing to switch to.

Below it, **Display Mode** — **Fullscreen** or **Free**. Fullscreen fits the layer to the canvas and works on *every* source type, webcam, text, and image included; Free makes it draggable and resizable. Switching to Fullscreen and back restores your free-mode position and size exactly, and the Transform section hides itself while Fullscreen is active (its sliders would do nothing). Per-type extras — text editing, waveform time window and so on — sit in the same section.

- **Keying** (webcam & image layers) — **Person Segmentation** (Apple Vision background removal) with **Fast / Balanced / Accurate** quality (default is Balanced on Apple Silicon, Fast on Intel).
- **Audio Reactivity** — the reactive source picker + **Sensitivity** (0–2.0; 1.0 = normal). See §12.
- **FX Chain** — this layer's effect stack: add, remove, **drag to reorder**, open each effect's editor. With multiple layers selected, chain edits apply to all. If you've MIDI-learned an effect's bypass toggle, **the mapping follows that effect** when you reorder the chain or delete something above it — your controller keeps hitting the effect you mapped, not whatever landed in that slot.
- **Compositing** — **Opacity**; **Blend mode** (Normal, Screen, Add, Multiply, Overlay, Soft Light, Hard Light, Color Dodge, Color Burn, Difference, Exclusion); **FX Only**; **Render through post-FX** (whether the global FX chain applies to this layer — "Render above global effects" in the right-click menu is the same switch flipped).
- **Transform** — numeric X/Y, W/H with an **aspect-lock padlock**, Rotation ($\pm 180^\circ$), Flip H/V, and Reset.
- **Performance** — per-layer **Render Scale** (0.25–1.0): render a heavy generator at lower internal resolution while the rest stays sharp.

Every properties slider supports **right-click** → **MIDI Learn** (§16).

Hand tracking

Orra Visualizer tracks your hands through the webcam — **both hands, the full 21-joint skeleton** (every fingertip, knuckle, and wrist), using Apple's Vision framework, entirely on your Mac.

There's no switch to find: **add a hand-tracking effect to a webcam layer and tracking turns on by itself**. Effects can follow your fingertips, react to a pinch, wrap visuals around your hands — whatever the plugin does with the joints it reads. Move your hand; the visual follows, live.

For plugin authors this is a first-class surface: the entire joint atlas is exposed to shaders (with tuned smoothing on the key points), so you can write effects driven by any part of either hand — the in-app templates document the whole API. On Intel Macs, tracking is automatically rate-limited to protect frame rate.

One quirk to know: a *generative* layer can't keep a crop and a hand-tracking effect at the same time (adding the effect resets the crop). Webcam and image layers are unaffected.

Face tracking

Face tracking works exactly the same way, on the same Vision framework, on your Mac: **add a face-filter effect to a webcam layer and it turns itself on**. Nothing to enable.

The surface is deep — **71 points across up to two faces at once**, covering the face contour, both eyebrows, both eyes and their pupils, the nose crest, inner and outer lips, and the overall face box. That's what lets a filter enlarge an eye, track a mouth, put something on your forehead, or paint a cheek and have it stay put while you move.

Face filters *are* effects — put one on the webcam layer and it rewrites that layer's image. They're audio-reactive too: the face drives *placement* while the music drives *intensity*, and a plugin can read both surfaces in the same shader.

Hands and faces are demanded independently, so a face filter doesn't switch on hand tracking (or pay for it) and vice versa — and a plugin that wants both can have both.

9. The mixer & audio sources

Open with **Cmd+M**. Every audio source is a channel strip; the **MASTER** strip on the right controls the summed mix. Drag the panel's top edge to resize it.

A channel strip

- **Color bar** (top) — gold, or the DAW track color for Orra Link strips.
- **Transport button + arm dot** — on controllable sources only (§11).
- **Stereo meters** with dB scale (–60 to +6; red above 0 dB) and peak-hold.
- **Fader** — professional IEC taper with +12 dB headroom. **Double-click resets to 0 dB**. Click the dB readout to *type* an exact value (–6, +3, –inf).
- **S / M** — solo / mute.

Right-click a strip: Change Source (rebinds in place — layers referencing the channel follow), **Channel Mode** (Stereo/**Mono** — mono grabs one hardware input and feeds both sides, ideal for a single mic on an interface), **Input Channel** (which hardware channels to take), Effects..., Rename..., Remove.

Channel effects — the FX button

Every strip — including **MASTER** — has an **FX** button that opens its insert chain: a stack of studio effects processed in order, top to bottom, before the fader. Seven effects to build from:

- **Gain** — clean trim, ±30 dB.
- **EQ** — 4-band parametric: low-cut filter, low shelf, mid bell, high shelf.
- **Saturation** — tape, tube, or clip warmth, with drive, mix, and automatic loudness matching.
- **Gate** — silences the channel below a threshold (kill room noise on a mic).
- **Compressor** — evens out dynamics; threshold, ratio, attack, release.
- **Duck** — drops this channel whenever *another* channel speaks: pick a **Key** strip and set the depth. Put a Duck on your music channel, key it to your mic, and the music dips while you talk — the classic radio move.
- **Limiter** — a ceiling; put one on MASTER so nothing ever clips.

Add, reorder, bypass (the ● eye), or remove effects live while audio plays; the button lights gold when a chain is active. Double-click any knob to reset it. **What you hear is what everything gets** — the processed signal drives the strip's meter, the visuals of every layer listening to that channel, the master mix, and recordings identically. Gate your mic and its layers stop twitching on room noise; that's the point.

Chains save with your project.

Adding sources — the + button

Source	What you get	Permission
System Audio → Whole System	Everything the Mac plays	Screen Recording
System Audio → Application	One app, isolated (Spotify, your DAW, a browser...)	Screen Recording
Input Device	Any Core Audio input — mics, interfaces, BlackHole, aggregates	Microphone (for mics)
Orra Link	A DAW track via the Orra Link plugin (§10)	none

Notes worth knowing:

- The Application list shows apps that can actually deliver audio (real apps, currently playing or known media players). Voice apps that hold audio without producing capturable output (Discord, Messages, Claude) are excluded on purpose.
 - Per-app capture **self-heals**: quit and relaunch Spotify and its strip reconnects within a few seconds automatically.
 - macOS applies a *fresh* Screen Recording grant only after relaunching the app — Orra Visualizer walks you through it the first time.
-

10. Orra Link

Orra Link is Orra Audio's DAW bridge: drop the Link plugin on any track in your DAW and that track appears in Orra Visualizer — no routing, no virtual drivers.

- **Auto-add:** put Link on a track while Orra Visualizer runs and a mixer strip appears by itself, carrying the **track's name and color** (it keeps following the DAW until you rename the strip yourself).
 - **Manual slots:** 36 pre-bindable slots let you wire channels before the DAW is even open.
 - **Rescan: Cmd+L** (or the + menu → Scan for Orra Link) surfaces any claimed channels that aren't in the mixer yet — including ones from before Orra Visualizer launched.
 - **Link strips are reactivity-only:** they carry audio for meters and visuals but no play/stop controls. A DAW's transport lives on its app-capture strip (§11) — one DAW usually feeds many Link tracks, and they'd all fight over one transport.
 - **Per-track spectral view for plugins:** Link strips also feed plugins a frequency-content view of every Link track at once, addressable by track name — so one visual can respond to the kick, the vocal, and the hats as separate instruments.
-

11. Transport control

Control playback of your music apps and your DAW **from inside Orra Visualizer** — per strip buttons, one master key, no drivers. (Full setup walkthroughs per DAW: `docs/TRANSPORT_SETUP.md`.)

The concept

Controllable strips grow two header controls:

- **▶ / ■ button** — play/stop *this* source now. **Option-click = exclusive solo**: this source plays, every other controllable source stops.
- **• arm dot** — gold = included when you press **L**.

L is the master transport key: one press starts every armed source, the next stops them all. It's deliberately *not* the spacebar, so your DAW's own space/play muscle memory never collides. Arm dots save with the project.

What's controllable

- **Music apps** — Spotify and Apple Music, zero setup: the first click asks macOS for permission ("Orra Visualizer wants to control Spotify") — once, ever.
- **DAWs** (captured as app-audio strips) — Ableton Live, Logic Pro, FL Studio, Cubase, Bitwig, Pro Tools, REAPER, Studio One, Digital Performer, Renoise, Reason, Ardour, Waveform, and LUNA are recognized automatically. **Any other DAW**: right-click its strip → **MIDI Transport (unlisted DAW)**.

How DAW control works

Each DAW gets its **own virtual MIDI port** — "Orra Visualizer Transport (Ableton Live)", "Orra Visualizer Transport (REAPER)" — so controlling one DAW never touches another. Logic, Cubase, and Pro Tools respond to machine control natively (enable MMC input, done). Everything else takes a one-time MIDI-map:

1. Enable the Orra Visualizer port in the DAW's MIDI settings (control/remote input).
2. Put the DAW in MIDI-learn on its **Play** control, then in Orra Visualizer **right-click the strip's transport button** → **Send Play**.
3. Repeat with **Stop** and **Send Stop**.

Send Play/Send Stop always transmit exactly what they say, so the mapping can never come out backwards. Two minutes, once, saved in your DAW's preferences.

Reading the icons

Apps report their true state (the icon is always live). DAWs can't report back, so a DAW strip's **▶/■ follows its own audio**: playing shows within a couple of seconds of sound arriving; it settles back to **▶** after ~2.5 s of silence. Pressing **L** or clicking the button updates it instantly either way.

12. Audio reactivity

Orra Visualizer continuously analyzes your audio: **bass** (20–200 Hz), **mids** (200–2k), **highs** (2k–20k), overall loudness, **beat detection**, the raw spectrum, and the waveform itself. Plugins drink from this — that's what makes everything move.

What a plugin can hear runs deep. Every plugin gets the full 1024-band spectrum — studio-analyzer resolution, fine enough to watch the kick drum and the hi-hats as separate signals — plus the raw stereo waveform itself. And when Orra Link tracks are in the mixer, a plugin can read a per-track spectral view of every Link-bound DAW track at once, addressable by track name. Beat detection and the bass/mid/high bands ride on top.

Choose what each layer listens to. Click the **headphones icon** on a layer row (or the Audio Source button in its properties):

- **Master (Mix)** — the summed mixer output (default).
- **Any mixer channel** — this layer reacts only to, say, the kick bus.
- **Any Orra Link channel** — react to one DAW track directly.
- **None** — the layer goes static.

Sensitivity (0–2.0) scales how hard the layer reacts. Frequency-band routing (which band drives which parameter) lives inside each plugin's editor — most factory plugins expose a band selector and per-band controls.

Global FX have the same per-effect source picker (§17) — a scene-wide effect can pump from a different channel than the layers do.

13. Plugins

Plugins are single **JavaScript files** (`.js`) carrying GPU shaders. Six kinds: **generators** (make visuals), **particles** (GPU particle systems, up to a million particles), **3D** (real mesh geometry with a vertex shader), **effects** (process the layer below), **transitions** (scene switches), and **composite FX** (process the whole mix). **Face filters** and **hand-tracking** plugins are effects too — they read the camera's landmarks and rewrite the layer beneath, which is why they go on a webcam layer (§8). Orra Visualizer also imports **ISF** (`.fs` / `.frag`) and **Shadertoy** (`.shadertoy` / `.toy`) shaders directly — put the file in the Plugins folder (View → Open Plugins Folder) and it loads in a second. What to know:

- **ISF** (editor.isf.video offers direct downloads): the download is a *folder* — take the inner `.fs` file out of it. Single-file shaders work; ones that ship a custom vertex shader (`.vs` sidecar) aren't supported and say so. Shaders with an image input file as effects; the rest as generators.
- **Shadertoy** (copy-paste): paste the **Image** tab's code into a plain-text file ending in `.toy`. **If the shader has a "Common" tab, paste its contents above the Image code in the same file** — that's where shared functions live, and skipping it causes "undeclared identifier" errors. Shaders with Buffer A/B tabs (multi-pass) aren't supported. Bonus: `iChannel1` maps to the live audio spectrum, so Shadertoy music visualizers react to your mixer.
- **Licensing**: Shadertoy shaders default to a non-commercial license (CC BY-NC-SA) unless the author says otherwise — check before using an imported shader in commercial work. Much of the ISF library is permissively licensed.

The Plugins folder

`~/Library/Application Support/OrrViz/Plugins/` — open it any time with **View** → **Open Plugins Folder**. Four homes: `FX`, `Generative`, `Particles`, `Transitions`. Subfolders you create become submenus (use **New Folder** right in the add-menus). Plugins that don't declare a folder land in each home's `Unsorted/`.

One level of grouping is what the menus read, so `Generative/Blobs` becomes a **Blobs** submenu — but `Generative/Blobs/Extras` would collapse back to `Blobs`. Keep folders one deep.

Packs

A **pack** (`.orrviz-pack`) is a bundle of plugins — and optionally scenes — in a single file. It's how content gets shared and how collections arrive.

- **Installing**: drop the pack onto the window. Its folder structure is preserved, so a pack that groups its plugins under `Blobs` gives you a `Blobs` submenu under the right type home. Anything uncategorised lands in `Unsorted/`. Name clashes are resolved quietly by numbering the newcomer, with its display name kept in step.
- **Exporting**: **File** → **Export Pack**. The dialog opens **collapsed** — four section headers with counts, an inventory rather than a wall — and you expand what you actually mean to include. Clicking a row's checkbox toggles that whole subtree; clicking further right just expands or collapses it.
- **Scene packs**: a pack can carry scenes. Dropping one **installs its plugins first**, then — if it holds exactly one scene — offers to load that scene into your current slot. If that slot already has layers on it, you get a second confirmation (**Load Scene / Keep Current**, defaulting to keeping what you have) because replacing a live scene can't be undone. The pack is installed either way, and multi-scene packs skip the offer entirely — load those from the **Presets** tab.

Sharing — plugins, packs, scenes, themes

Everything you make in Orra Visualizer travels as a file, and everything installs by dropping that file on the window. The four shapes:

- A **plugin** is one `.js` file. Send it as-is; the receiver drops it on their canvas and it's installed (**Installing a plugin** below).
- A **pack** (`.orrviz-pack`) is a collection — plugins, and optionally scenes — exported from **File** → **Export Pack**. Folder structure survives the trip, so your organisation arrives intact on the other side.

- A **scene** exports from the scene chip (right-click → **Share Scene**) or **File** → **Export Scene**: a one-scene pack carrying the layers, effects, parameter settings, and every plugin the scene uses. The receiver drops it, the plugins install, and they're offered the scene into their current slot.
- A **theme** exports from **Display** → **Appearance** → **Export Theme** as a small `.json`; dropping one on any copy of the app installs it and applies it instantly (§20).

There's no account, no store, no upload step — a Discord message or an AirDrop is the whole distribution pipeline.

Installing a plugin

Drop the `.js` file straight onto the canvas. Easiest install there is — made for beginners: drag a plugin file from Finder (or a download, or one a friend sent you) onto the Orra Visualizer window and let go. That's it:

- It's validated first — a broken file never installs; a toast tells you the exact parse error.
- It files itself into the right folder by its declared category.
- Generators/particles **spawn as a layer immediately**; effects apply to the selected layer; transitions point you to the scene transition menu.
- **Name collision?** A modal offers **Overwrite / Rename / Cancel** — Rename suggests a free name and projects whatever you type into the plugin's display name, so menus always match.

Live editing

Plugins **hot-reload**: save the file (from Orra Visualizer's code editor or any external editor) and running layers update within half a second. A plugin's **custom UI window doesn't refresh in place** — close and reopen it (**P** is the shortcut) to see UI changes. Editing a plugin makes it yours — updates will never overwrite your edits.

The Library (Cmd+B)

A searchable browser over everything installed: **Sources** (generators, particles, 3D), **Effects**, and **Presets** tabs, with live search across names, tags, authors, and folders. Add via the button, double-click, or Return. **Cmd+B** again closes it.

14. The plugin editor

Open a layer's plugin editor with **P** (or the pencil icon). It opens bottom-right, over the panel side, so the canvas stays watchable.

- **Preset bar** — save, load, and delete **per-plugin presets**. Everything the plugin exposes (params + reactivity routing) is captured.
- **Parameter undo/redo** — the « » arrows, or **Cmd+Z / Cmd+Shift+Z** while the editor is focused. One knob-drag = one step. Loading a preset is one step too — **audition a preset, hate it, Cmd+Z back to your sound**. This history is the editor's own; your canvas undo is untouched.
- **Controls** — factory-style gold knobs (double-click any knob to reset), or full custom web UIs on plugins that ship one. Band selectors route which frequency range drives the plugin.
- **MIDI Learn** — right-click any knob → **MIDI Learn**, twist a controller knob, done. The menu shows the current mapping and offers **Clear**. (§16)
- **Edit Code** — opens the plugin's source in the code editor.
- **Headless mode** — closing the window of a live-running plugin *parks* it: the UI disappears but the plugin keeps animating. Press **P** to bring the window back. (Static plugins simply close.)
- The window scales as one piece — drag a corner and the whole UI grows.

15. Creating your own plugins

Cmd+Shift+C (or **Create New** in any add-menu) opens the **Plugin Creator** — a code editor pre-loaded with a complete, heavily-commented **mega-template** for each type. Six tabs, each keeping its own draft:

Generator · Particles · FX · Transition · Face Filter · Hand Tracking

Cmd+Shift+F opens the same window on the **FX** tab. Face Filter and Hand Tracking have no shortcut of their own — click their tabs to reach them.

- Pick a **folder**, name the file, edit, **Save** — the shader compiles, and on success the plugin drops onto your canvas as a layer, live.
- Saved plugins hot-reload like any other; keep the editor open and iterate against the live canvas.
- **Editing** an existing plugin opens the same window without the tab strip, and adds a **folder button** showing where the plugin currently lives (e.g. "Generative / 3D"). Click it to move the file to another folder within the same type home, or to make a new one.

The templates document the entire surface — audio uniforms, the full 1024-band spectrum, the raw waveform, per-track Orra Link bands, beat detection, multi-pass rendering and feedback buffers, custom web UIs, dynamic textures (push images or text from the plugin's own UI into the shader), particle simulation, 3D meshes, and the hand and face landmark surfaces.

Creating with an AI

You don't need to know GLSL to make plugins — the Plugin Creator was built for a conversation between you, the canvas, and the AI of your choice (Claude, ChatGPT, anything with a chat box).

Making something new:

1. Open the Creator (**Cmd+Shift+C**) on the tab for what you want (Generator, Particles, FX, Transition, Face Filter, or Hand Tracking).
2. Get the template to your AI: **Copy All** puts the entire editor — framework comments and code — on the clipboard, or use **Drag .txt** to drag it into the chat as a file (some AIs prefer an attachment over a very long paste).
3. Tell the AI what you want, in plain language: *"make the particles form a spiral galaxy that tightens on the bass and flashes on every beat."* The template it just read is the contract — it tells the AI exactly which audio uniforms exist and how the plugin must be shaped.
4. Paste the AI's file back into the editor and **Save**. It's on your canvas, live, reacting to your music.

When it doesn't compile — the Copy Error button:

A failed save keeps the editor open with the compiler's output in a red error pane — and a **Copy Error** button, which only appears when there's an error to copy. One click puts a complete, ready-to-paste bundle on your clipboard:

- a fix-this instruction (return the corrected file, keep the plugin's name and identity exactly as they are),
- the **exact compiler error**,
- a **reference for what Orra Visualizer actually feeds shaders** — matched to your plugin's type, so the AI can't "fix" your code with Shadertoy names or invented capabilities that would compile but never animate,
- and your **complete plugin source**.

Paste the bundle into any chat AI, paste its corrected file back over your editor contents, Save. Most errors round-trip in one pass.

If the AI keeps missing, give it the source of truth: tell it to refer to the matching **mega-template** — Generator, Particles, FX, Transition, Face Filter, or Hand Tracking — and hand that template over (open its tab in the Creator, **Copy All** or **Drag .txt** into the same chat). The templates are the complete, commented contract for each plugin type; with one in context, the AI stops guessing about what the engine supports.

The loop. Because saves hot-reload in under a second, iteration is conversational: *"more glow"* → *paste* → *save* → *watch* → *"slower rotation, react to highs instead"* → *paste* → *save*. Keep the editor beside the canvas and sculpt the visual by prompt. When you land somewhere good, it's already a real plugin — file it in a folder, map its knobs to MIDI, save presets, share the `.js`.

16. MIDI control

Any plugin parameter and any layer-properties slider can be driven by a MIDI controller:

1. **Right-click** the knob/slider → **MIDI Learn**.
2. Move a control on your MIDI device. Mapped.
3. Right-click again to see the mapping ("CC 21 (Ch 1)") or **Clear** it.

Buttons are learnable too — a layer row's **S** (solo) and **M** (mute) buttons both take a right-click → MIDI Learn, as do the bypass toggles on a layer's FX chain. Right-clicking one of these opens the learn menu instead of firing the button, so you can't toggle a layer by accident while mapping.

Two places MIDI Learn deliberately isn't offered: the **transition editor** (§4) and the **Global FX** slots — in both cases the parameters sit outside the layer-keyed mapping model, so rather than let you create a binding that silently wouldn't work, the menu simply doesn't appear.

Mappings save with the project, and **a mapped control stays live no matter which scene you're looking at** — trigger a control you learned on scene 1 while scene 2 is up and it still drives its scene-1 target. Treat your controller as one layout for the whole project. The status bar's **MIDI** dot flashes on activity.

(The "Orra Visualizer Transport" MIDI ports in §11 are outputs *to* your DAW; MIDI Learn listens to inputs *from* your controllers.)

17. Global FX

The **Global FX** tab holds a chain of effects applied to the **entire composited output** — color grades, glitch passes, film looks over everything at once.

- **+ Add Effect** (or **Create FX** to write one). Drag slots to reorder.
- Each slot: **enable/bypass** (click the badge), **Solo** (isolate it in the chain), **Mute**, **edit** (pencil or double-click), **remove** (x), and a **headphones** source picker — each global effect can react to its own audio source.
- Per-layer opt-out: a layer with **"Render above global effects"** set (or "Render through post-FX" off) floats above the global chain — keep a webcam clean while the world behind it melts.

Global FX are per-scene, like layers.

18. Recording

- **R** (or the **REC** button) starts and stops. While recording the button reads **STOP 0:00** with a live counter.
- Files land on your **Desktop** by default (configurable), named `OrrViz_YYYYMMDD_HHMMSS` — with a `.mov` extension on HEVC, `.mp4` on H.264. If your chosen folder has gone missing, it falls back to the Desktop.
- **File** → **Recording Settings**: Quality (Low / Medium / High / Maximum), Codec (**HEVC (smaller files)** / **H.264 (compatible)**), Frame rate (**Auto (follow display)** / 30 fps / 60 fps), and the save folder — with a live bitrate/size estimate.
- **Resolution**: recordings render at the project's **output resolution** (set at New Project or in the Disp tab — 4K default for unnamed projects), independent of window size or render scaling. The preview can run scaled; the recording is always full-res.
- While recording, output size and window resizing **lock** (so the encoder can't desync) and fullscreen toggling is blocked.
- **View** → **Record App Window (Demo)** records the entire app window with UI and cursor — for tutorials.

Recorded audio is the full time-aligned mixer mix (every channel post-fader, plus mic). Sources are aligned on a common clock — multi-source recordings stay in sync even when macOS delivers their audio at different latencies.

19. Streaming

Click **LIVE**:

- **Twitch / YouTube** — click **Connect**, approve in the browser, and the RTMP URL and stream key fill themselves. The dialog remembers you ("Connected as ...").
 - **Custom RTMP** — paste any URL + key.
 - **Go Live** streams at 1920×1080 @ 30 fps, 4500 kbps video and 160 kbps audio — squarely in Twitch's recommended range for 1080p30. The button turns green and reads **END** while live; the status bar's **LIVE** dot lights.
-

20. Display & performance

The **Disp** tab is the control room, in four collapsible sections: **Appearance**, **Display**, **Grid**, and **Shortcuts** (a read-only cheat sheet, collapsed by default).

Display

- **FPS Cap** — 30 fps / 60 fps / 120 fps / **Unlimited**. Also on View → FPS Cap, where the last option reads *Uncapped*.
- **Render Resolution** — **Auto** / **100%** / **75%** / **50%** / **25%**. Auto watches your frame rate and adapts the internal resolution automatically (a high-quality bicubic upscale keeps it looking clean); the fixed percentages switch Auto off and pin the scale. Recordings are unaffected — this is preview-only.
- **Always On Top** — the same switch as View → Always On Top.
- **Optimize Performance** — the project-wide adaptive performance toggle.
- **Window Size** — **Free Resize**, a list of named presets, or **Custom...** (which reveals a W × H pair; minimum 100 × 100, applied on Return). The presets name their platform, so you pick by destination rather than by arithmetic: 1920×1080 for YouTube/Twitch/Kick, 2560×1440 for YouTube 2K, 3840×2160 for 4K, 1280×720 for 720p, **1080×1920 for TikTok/Reels/Shorts**, 2160×3840 for 4K vertical, 1080×1350 for Instagram and Facebook feed, 1080×1080 for square feed posts, and 1000×1500 for Pinterest pins. Selecting a preset locks the canvas aspect while you resize the window.
- **Safe Zones** — broadcast-safe guides.

Grid

Grid and **Snap** toggles, a **Density** slider (2–100, default 10), and a **Ghost** slider (0–1, default 0.2) for how strongly the grid shows.

Appearance — themes

Orra Visualizer's entire interface is themed at runtime, and you can build your own.

- **Theme** — the picker. Nine themes ship with the app: **Default**, **Classic 95'**, **Classic Grey / Black**, **Grey / Black**, **Grey / Green**, **Grey / Red**, **Classic Dark**, **Slate**, and **Light**. New installs start on **Classic Dark**. Any themes you've saved or imported appear below a divider.
- **Shell** and **Accent** swatches — the two colours everything else is derived from. Drag either one and the interface recolours live; starting from a built-in is the intended way to begin, and the first drag converts it into a **Custom** palette seeded from wherever you started.
- **Save Theme** names and keeps the current custom palette (enabled once you've made one). **Delete** removes a saved theme — the colours stay applied, they just lose their name. **Export Theme** writes a `.json`, defaulting to your Desktop.

Sharing themes is a drag and a drop. Drop any theme `.json` onto the window and it installs *and* applies instantly — no dialog, no relaunch, just a "Theme applied" toast. It persists across launches from that moment. If a theme arrives with the same name as one of the built-ins it's kept separately as an import rather than overwriting anything.

Themes are also switchable from **View** → **Theme**.

Elsewhere

Per-layer: the **Performance** → **Render Scale** slider (§8) lets one heavy generator render lighter without touching anything else.

The menu-bar readout shows live FPS — and the effective render scale (`75% · 60`) whenever scaling is active.

21. Keyboard shortcuts — complete reference

One-key (canvas focused)

Key	Action
S / Tab	Toggle side panels
R	Start/stop recording
L	Master transport — play/stop all armed sources
M	Mute selected layer
P	Toggle plugin editor for selected layer
G / Shift+G	Toggle grid / grid snap
1–9, 0	Switch to scene 1–10
Arrows / Shift+Arrows	Nudge selected layers 1 px / 10 px
Delete or Backspace	Delete selected layer(s)

These are the plain, unmodified keys — hold Shift or Option and they do nothing, so they won't fire by accident while you're typing elsewhere.

With the mouse

Combo	Action
Cmd+drag corner handle	Rotate layer
Shift/Cmd+click	Multi-select layers
Shift+C+click	Center the clicked layer
Option+scroll	Scale selected layer
Option+click transport ►	Solo that audio source
Drag scene square	Swap two scenes
Option+drag scene square	Copy scene
Cmd+click scene square	Open that scene's transition editor
Cmd+scroll over project grid	Resize project thumbnails
Right-click knob/slider	MIDI Learn

Command keys

Combo	Action
Cmd+Z / Cmd+Shift+Z	Undo / Redo (canvas; parameter undo inside a focused plugin editor)
Cmd+C / Cmd+V / Cmd+D	Copy / paste / duplicate layer
Cmd+A / Cmd+Shift+A	Select all layers / deselect all

Cmd+B	Library browser (toggle)
Cmd+Shift+C	Plugin Creator — Generative tab
Cmd+Shift+F	Plugin Creator — FX tab
Cmd+M	Toggle mixer
Cmd+L	Rescan for Orra Link
Cmd+N / Cmd+O	New / Open project
Cmd+S / Cmd+Shift+S	Save / Save As
Cmd+Shift+O	Project Browser
Cmd+Shift+T	Capture thumbnail
Cmd+W	Close project (app stays open)
Ctrl+Cmd+F	Performance View (fullscreen canvas)

Contextual

Key	Context	Action
Enter / Esc / Backspace	Crop lasso	Commit / cancel / remove last point
Enter / Esc	Canvas text editing	Commit / cancel
Esc	Eyedropper	Cancel sampling

22. macOS permissions

Orra Visualizer asks only when a feature needs it. If you ever denied a prompt (or never saw one), **Help** → **Permissions** shows every permission with a checkmark when granted and a one-click fix when not.

Permission	Needed for	Notes
Camera	Webcam layers	
Microphone	Mic input sources	
Screen Recording	Screen-capture layers and system-audio capture (macOS files app audio under Screen Recording)	A fresh grant applies only after relaunching Orra Visualizer — the app walks you through it.
Automation (App Control)	Play/pausing Spotify & Apple Music	One prompt per app, ever.

23. Licensing

- **3-day free demo** — one per machine, starts from the launch gate.
 - **Activation** — enter your license key (from your purchase email); licenses allow **3 machine activations**. Free a seat any time with **Deactivate** in Help → **License**, which also shows your status, activations, and support links.
 - **Offline-friendly** — once activated, Orra Visualizer works forever offline; it never phones home to keep running.
-

24. Your files & backups

Everything lives under `~/Library/Application Support/0rrViz/`:

Folder / file	Contents	Back up?
<code>Projects/</code>	Your <code>.ovz</code> projects	Yes
<code>Plugins/</code>	Every plugin you've installed or written	Yes — your authored work
<code>Presets/</code> , <code>PluginPresets/</code>	Scene/layer and plugin presets	Yes
<code>Themes/</code>	Themes you've saved or imported	Yes
<code>Assets/</code>	Imported assets	Yes
<code>Preferences/</code>	Display/audio settings, theme, tooltip language, recent files	Optional
<code>CrashLogs/</code>	Diagnostics	No

The app log (for support) is `~/Library/Logs/0rrViz/0rrviz.log`.

25. Troubleshooting

Symptom	Fix
Shortcuts do nothing	Something else has keyboard focus — click the canvas once.
Add Generative / Add Particles menus are empty	Nothing's installed yet. Drop a <code>.js</code> plugin or an <code>.orrviz-pack</code> onto the window — see §13.
The Transition menu only offers None and Default	Same cause — those two are built in; every other transition is a plugin you install.
Cmd+click on a scene square does nothing	That scene's transition is set to None or Default, which have no parameters to edit. Pick a transition plugin first.
A layer's MIDI knob also moves a second layer	You duplicated the layer, and the copy inherited its MIDI bindings by design. Re-learn the control on one of them to split them.
System-audio source is silent, no prompt appeared	Help → Permissions → Screen & System Audio; after granting, relaunch (macOS applies it then).
Spotify strip went quiet after Spotify restarted	Wait a few seconds — capture self-heals.
The DAW ignores L	Its port isn't enabled for control input, or Play/Stop weren't both mapped — see <code>docs/TRANSPORT_SETUP.md</code> , and use the right-click Send Play / Send Stop helpers.
Transport mapping stopped working after relaunching Orra Visualizer	The DAW's MIDI device list went stale — rescan MIDI devices in the DAW (or restart it with Orra Visualizer running).
DAW strip shows ► while playing	It's outputting pure silence — the icon follows the strip's audio. One press of L re-converges.
A dropped plugin didn't show where expected	Check its <code>category:</code> — a typo (e.g. "generater") files it under FX. The plugin editor shows a warning strip for this.
An app won't screen-capture	It has no open window — Application capture grabs a window, not a hidden process.
Choppy preview on a 4K/5K display	Disp → Render Resolution → Auto (recordings stay full quality).
Everything is slow with one heavy layer	That layer's Performance → Render Scale slider.
The demo says it expired early	The demo clock is tamper-proof — changing the system clock doesn't extend it.