

Visionmixer Operator Manual

Version: 1.1.3 — built 2026-10-12

NDI Vision Mixer — software M/E vision switcher with integrated broadcast audio mixer, in the style of a Sony BVS-3200CP. Every video and audio input and output is NDI.

Visionmixer Operator Manual	1
About this manual	3
Quick Start	3
Install and launch the mixer	3
Put your first picture on PGM	4
Where to see the output	5
Before the show: the 10-second sanity check	5
Surface Tour	5
The header band	6
The left column	7
The right column	8
Colour language (broadcast standard)	8
Transitions	9
PGM and PST — two independent buses	9
CUT and AUTO — arm, then fire	9
MIX, WIPE, DME, STINGER — the type row	9
T-bar — the manual throw	10
RATE	10
KEY TAKE — direct per-key controls	10
FTB — FADE TO BLACK	10
Keyers	11
Fill source and mode	11
KEY CUT and KEY MIX — the everyday controls	11
Next-transition arming	11
Alpha interpretation — STRAIGHT or PREMUL	11
KEY ADJUST — the per-key DVE	12
Setting up LUMA and CHROMA keys without a rig	16
ME1 and E-MEM	16
The amber ME1 page	16
MASTER KEYS ACTIVE — the red lamp	17
The EFFECT BANK keypad	17
Store a look (the everyday E-MEM flow)	18

Recall a look	18
HOLD SRC — protect sources across recalls	18
Macros	18
The bank	20
Record a macro	20
Replay a macro	20
Edit a macro	20
Conditions	21
Fader memory (per-strip audio snapshots)	22
Audio	22
The sixteen strips	22
Fader, mute, solo, pan	23
EQ and dynamics	23
Lipsync	23
Metering — EBU	23
The monitor section	24
Test tone and program audio on PGM	24
Media, Stinger, Caption and OGraf	24
The media player	24
Stinger transitions	26
Captions	26
OGraf graphics	27
Still Store	28
The pool	28
The POOL browser — click to swap	29
Taking a still to air	29
Formats and alpha keying	30
Persistence	30
Sessions	30
Save the current state	31
Load a session	31
Delete a session	32
START FRESH — clean start from scratch	32
When sessions are your friend	32
Tools and Utilities	32
Test patterns	32
The ROUTER page	33

Multiviewer	33
HDMI clean output	33
PERF — the performance overlay	34
Frame presets preview	34
Remote Control (vmctl)	34
The control socket	35
Common examples	35
Operational notes	35
Installation and Operations	36
Requirements	36
Installing	36
Starting	37
Configuration	37
Updating	37
Uninstalling	37
Licensing	37
Troubleshooting	38
Symptom → fix	38
The 50 fps law	39
Glossary	39

About this manual

This manual is written for the **broadcast operator first**: the person sitting at the panel during a show. Installation and system operations are covered second, in the later chapters. Procedures are numbered step by step and use the **exact button names printed on the panel** (for example CUT, AUTO, BKGD, FTB, POOL...).

- Chapters 01–02 get you on air and show you around the surface.
- Chapters 03–09 are the show toolkit: transitions, keyers, ME1/E-MEM, macros, audio, media/graphics, and the still store.
- Chapters 10–12 cover sessions, utilities, and remote control.
- Chapters 13–14 cover installation, operations, and troubleshooting.

The two NDI outputs the mixer publishes are <machine> (PGM) (the on-air composite) and <machine> (Preset) (a true look-ahead preview of the next transition).

Quick Start

This chapter takes you from a fresh install to a picture on air, using only the mixer's built-in test signals — no NDI cameras required.

Install and launch the mixer

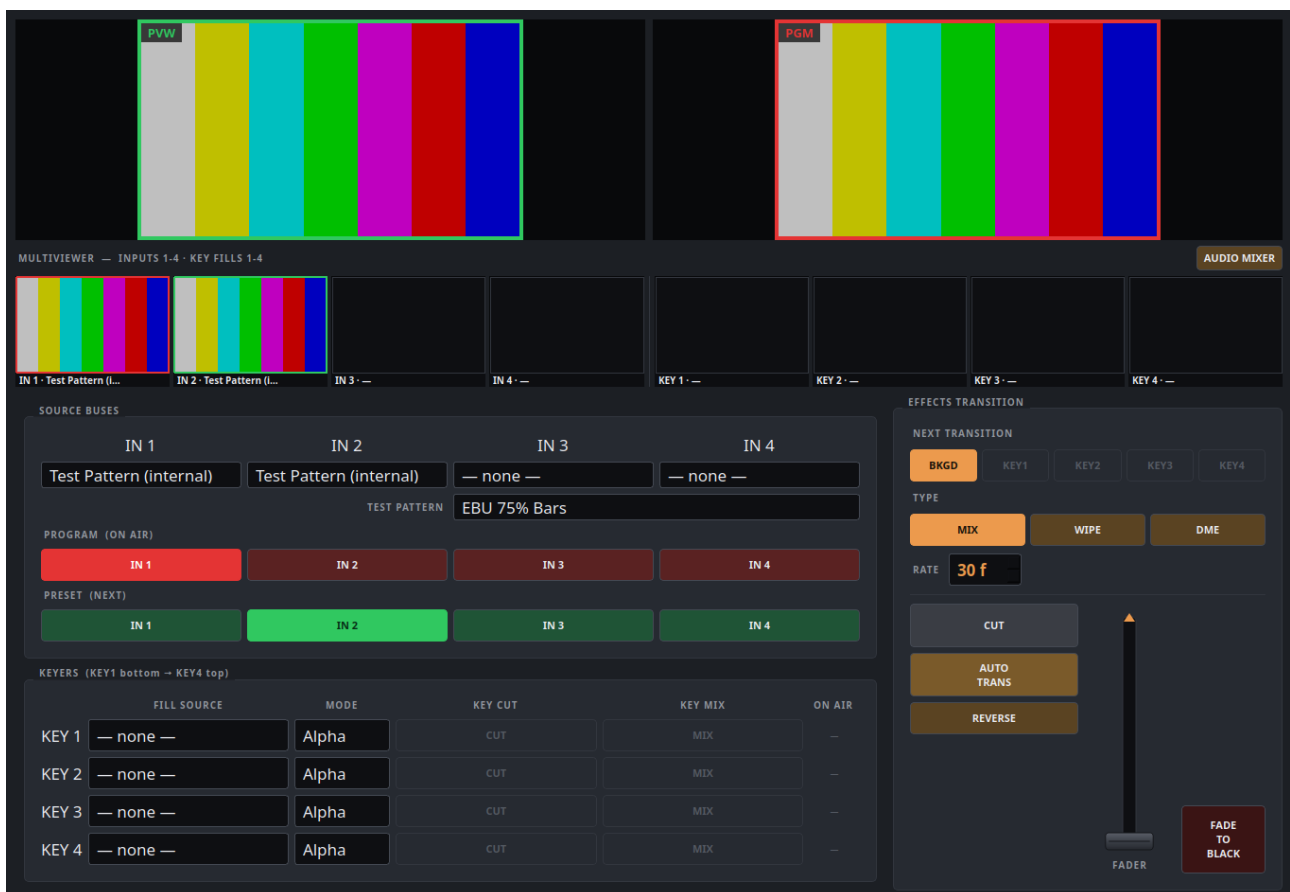
1. Install the package (see *Installation and Operations* for the full requirements).

Linux (Ubuntu 24.04 or newer):

```
sudo apt install ./visionmixer_<version>_amd64.deb  
(the file you downloaded, e.g. visionmixer_1.1.3_amd64.deb)
```

Windows (Windows 10 or 11, 64-bit): run `Visionmixer-setup-<version>.exe` and follow the Setup wizard. If Windows shows "Windows protected your PC" or an unknown-publisher warning, choose **More info**, then **Run anyway**.

2. Start **Vision Mixer** from the application menu (Linux; or run `visionmixer` in a terminal) or from the **Start menu** (Windows). The first launch begins a 30-day full-feature trial; enter your license key any time via the **LICENSE** button.
3. The control surface window opens with the **M/E** page showing. The header band reads **● LIVE**, the picture format (for example 1920×1080 · 50p) and the frame-rate selector.
4. If the *GPU capability* notice appears, tick *Don't show this again* and close it.



Put your first picture on PGM

With no cameras connected, use the internal test-signal generator — it is always available on every input dropdown:

1. In the **SOURCE BUSES** group (left column, M/E page), click the dropdown above **IN 1** and choose **Test Pattern (internal)**.
2. The **TEST PATTERN** dropdown appears in the group. Pick **EBU 75% Bars** (the default) — colour bars appear in the **IN 1** multiviewer tile.

3. Click the green **IN1** button in the **PRESET** row. The bar lights green: bars are now queued on preset (preview).
4. Press **BKGD** in the **EFFECTS TRANSITION** group (right column) so the amber BKGD lamp is lit — the next transition will act on the background.
5. Press **CUT** (the white button, bottom of the transition cluster). The buses flip-flop: bars are now on **PGM** (red monitor, right), and the PGM monitor shows them full size.

That is the whole switching model: *stage on PRESET, take with CUT or AUTO.*

Where to see the output

- **On the panel:** the **PGM** monitor (right-hand column, red border) shows the on-air composite; **PST** (green border, left of it) shows the look-ahead preview — what PGM will look like after your next transition.
- **On the network:** the mixer publishes two NDI sources any receiver can pull: <machine> (PGM) (the on-air composite, with program audio embedded) and <machine> (Preset) (the look-ahead preview, video only). Open them in an NDI multiviewer, OBS, or a hardware monitor.
- **On a projector or screen:** the **HDMI OUT** ■ button in the header gives a full-screen *clean* PGM (no UI, no tally) on the display you pick — see the Tools chapter.

Before the show: the 10-second sanity check

1. Header shows ● LIVE and the expected format.
2. Bars on PGM (procedure above) — proves the render loop is alive.
3. Press **FTB** once: the picture dissolves to black at the transition rate. Press **FTB** again to fade back up. Leave it up.
4. On the audio console (**AUDIO MIXER** button), the program meter idles near silence — see the Audio chapter for line-up.

Surface Tour

A map of the control surface. The panel is one window with a header band, a left column (multiviewer, crosspoints, media/still/graphics rows) and a right column (PST/PGM monitors, macros, transition + effect bank).



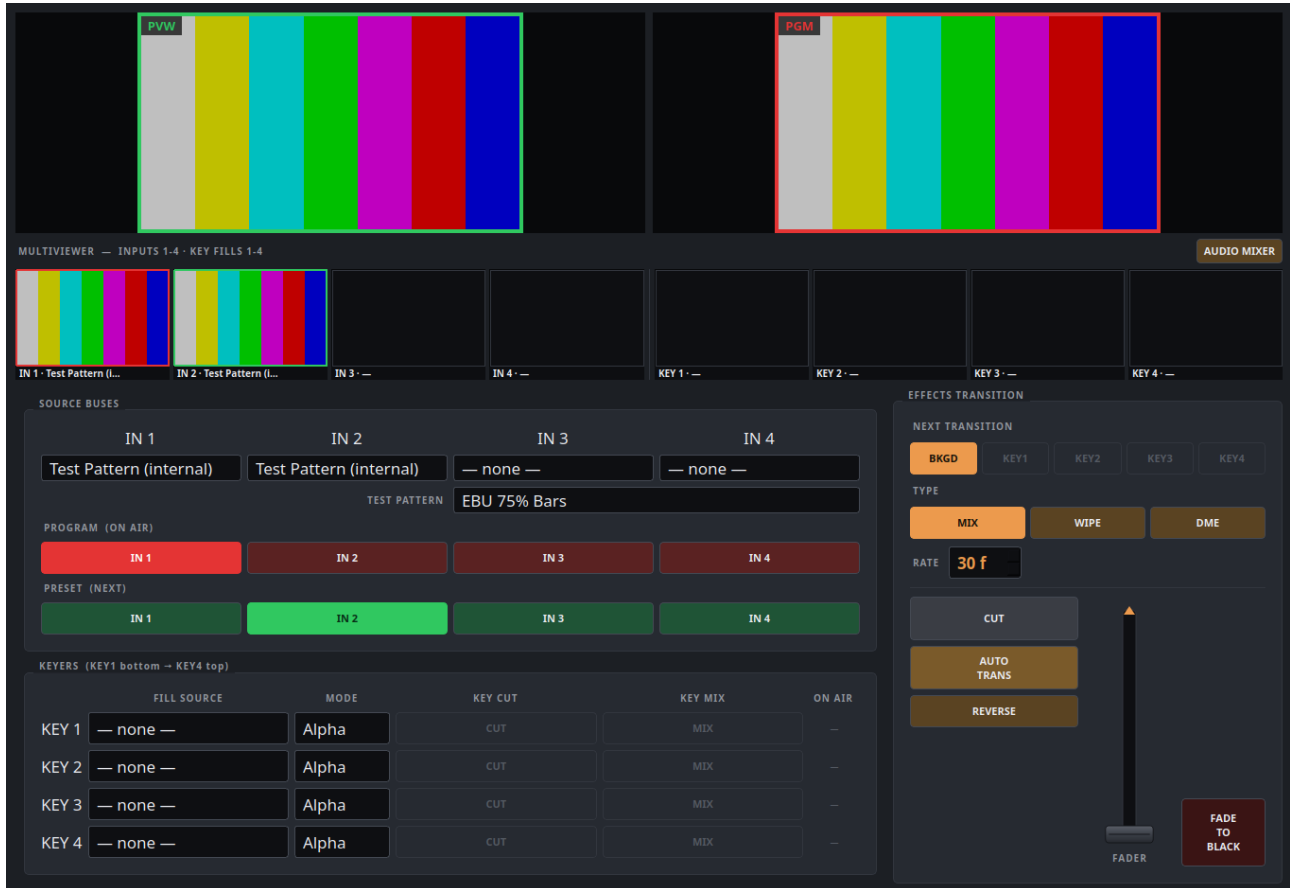
The header band

Across the top, left to right:

- **NDI VISION MIXER** title and the page navigation: **M/E**, **ME1**, **KEYS**, **ME1 KEYS**, **ROUTER**. The lit tab is the page you are on; the crosspoint area and the right-hand transition/effect stack change with it.
- **LIVE** lamp and the running format (for example 1920×1080 · 50p).
- **KEYERS n/K** — active keyers vs the GPU keyer budget (hidden when the engine does not report it).
- The **25p / 50p** frame-rate selector. A change is written to the config and applies on the next launch (· restart to apply hint when it differs from the running rate).
- **MEDIA** — shows/hides the floating media-player transport window.
- **SESSIONS** — opens the session save/load dialog (see the Sessions chapter).

- **PERF** — shows/hides the performance overlay (fps, render times).
- **HDMI OUT** ■ — clean full-screen PGM to a chosen display.
- **ROUTER** — jumps to the ROUTER page (routes the Preset NDI output only).

The left column



From the top:

1. **MULTIVIEWER — INPUTS 1-8 · ME1 · M/E**: eight input tiles in two rows, then a strip with the **ME1** tile and the **M/E** (master PGM) tile. The **AUDIO MIXER** button (amber) in the strip header opens the audio console.
2. The **crosspoint pages** (what changes per tab):
 - **M/E — SOURCE BUSES**: eight source dropdowns (IN 1...IN 8) plus the internal **ME1** and **BLK** columns; the **TEST PATTERN** dropdown appears when a slot uses the internal generator. Below them the red **PROGRAM** row (IN1–IN8, ME1, BLK) and the green **PRESET** row. Each row ends in a **HOLD SRC** lamp (recall-protect, see ME1 & E-MEM).
 - **ME1** — the amber second-M/E page (own chapter).
 - **KEYS** — the four keyer strips (own chapter).
 - **ME1 KEYS** — the same strips for ME1.
 - **ROUTER** — the output router matrix (Tools chapter).
3. The **transport row**: **MEDIA PLAYER** and **STILL STORE** group boxes side by side (own chapters).

4. The **compact row**: **AUDIO PLAYER**, **CAPTION** and **GRAPHICS (OGraf)** group boxes (Media chapter).

The right column

1. **PREVIEW / PROGRAM — MASTER**: the **PST** (green) and **PGM** (red) monitors. PST is the look-ahead preview — it shows what PGM will look like after the next transition, given the armed keys.
2. **MACROS**: sixteen macro buttons (M1...M16) with **REC** and **EDIT** modifiers (own chapter).
3. The per-page bottom row: **EFFECT BANK** keypad beside the **EFFECTS TRANSITION** cluster on the M/E and ME1 pages (KEYS pages point back to their M/E tab; ROUTER shows a note).



Colour language (broadcast standard)

- **Red** — program / on air (PROGRAM row, PGM monitor, on-air tallies).

- **Green** — preset / preview (PRESET row, PST monitor).
- **Amber** — armed / selected (transition arm BKGD/K1–K4, type buttons, latching modifiers).
- **Cyan** — selection / active chrome (page tabs, pickers, focus).
- **Greyed** — unassigned or unavailable (a keyer with no fill source, media without a GPU compositor).

Transitions

Everything about moving picture on and off air: the two buses, the arm, the fire buttons, the types, the T-bar, and **FTB**.

PGM and PST — two independent buses

- The **PROGRAM** row (red) selects what is on air *right now*. Punching a PROGRAM button is an instant hot-cut of the background — it never waits for CUT/AUTO and it never touches preset.
- The **PRESET** row (green) stages the *next* background. Nothing goes to air from preset until you fire a transition.
- The two rows are independent: after a take they **flip-flop** (the shot that was on air lands on preset), so repeated takes alternate between two shots — the classic way to cut camera-to-camera.
- On-air keys are untouched by a background hot-punch: they stay composited over whatever background you cut to.
- The internal columns punch like any input: **ME1** (the second M/E's re-entry) and **BLK** (black).

CUT and AUTO — arm, then fire

The next-transition arm (**NTS**) declares *what* the next take acts on:

- **BKGD** — the next transition swaps the PGM/PST backgrounds.
- **K1 ... K4** — the next transition *toggles* that key on/off air.

The arm is **sticky**: it stays lit after a transition (operators repeat the same take all show long). An empty arm (nothing lit) makes CUT/AUTO a no-op — safe, nothing happens.

Firing:

1. Arm what should move: press **BKGD** and/or **K1–K4** (amber lamps light).
2. Stage the shot on the **PRESET** row.
3. Fire:
 - **CUT** (white) — instant, on the next rendered frame. CUT ignores the transition type and the rate; there is no "wipe cut".
 - **AUTO** (amber) — a timed transition over the **RATE** (in frames). Pressing AUTO again mid-move snaps to the end; pressing CUT mid-move also jumps to the end state immediately.

All armed elements share one timeline: BKGD + K2 + K4 armed means the background moves while K2 and K4 toggle, all completing on the same frame.

MIX, WIPE, DME, STINGER — the type row

One type is lit at a time; it applies to the *background* of the next timed take (keys always dissolve — a key never wipes):

- **MIX** — cross-dissolve (the default).
- **WIPE** — sub-controls appear: **Pattern** (Horizontal, Vertical, Corner, Box), **Direction** (Forward/Reverse), **Border** (0 px = hard edge, up to 64 px border).
- **DME** — geometric moves: **DME Move** (Push, Squeeze) plus Direction.
- **STINGER** — a pre-imported animation take (see the Media chapter). The first STINGER click with nothing imported opens the import dialog; the loaded clip name and cut frame show under the type row (**IMPORT...**, **CUT POINT...**).
- **REV** — reverses the direction sense of the next throw (the wipe sweeps the other way) without changing buses.

T-bar — the manual throw

The vertical fader left of the KEY TAKE grid is the transition progress: all the way down is "not started", all the way up is "complete".

- Move it part-way and hold: the picture holds mid-transition (a held mix or a held wipe edge is a legal, held state — useful for split looks).
- Pull it back to the start: the transition scrubs back out and is abandoned — no swap, no key commit.
- Complete a full throw: buses swap, armed keys commit, and the bar's direction flips (the "bowtie") — the *next* take is performed by pushing the bar back the other way.
- Grab the bar during an AUTO: the motor releases and the move follows your hand; releasing mid-throw parks it there.

RATE

The **RATE** spinner (top right of the transition cluster) sets the AUTO duration in **frames** (1–999). Default is 25 f at 25p / 50 f at 50p (~1 s); changing the output frame rate rescales the live rate so the duration stays the same. A small seconds hint shows next to it. The rate is read at fire time — changing it mid-move does not affect the in-flight transition. FTB and KEY MIX fade at the same rate.

KEY TAKE — direct per-key controls

Next to the T-bar, the **KEY TAKE** grid gives every key a direct pair (see the Keyers chapter): a red **CUT** square (instant on/off) and an amber **MIX** square (dissolve at RATE), plus the per-key **HOLD SRC** lamp. These act on one key alone and never touch the arm.

FTB — FADE TO BLACK

The red **FTB** button (top right of the cluster) is the master output fade:

1. Press **FTB**: the whole PGM composite (background + all keys) dissolves to black at RATE and *latches* — it holds black.
2. Everything underneath keeps running: sources stay live, transitions still work under black ("re-rack under FTB"), so fading back up reveals the current live state.
3. Press **FTB** again to fade back up. A mid-ramp press reverses the ramp from wherever it is.
4. FTB changes no bus, key or arm state, and never affects the PST preview — you can set up the next shot while PGM is black.

Keyers

Four independent key layers, each with its own fill source and keyer mode. Keys stack **KEY1 (bottom)** → **KEY4 (top)**; higher numbers composite over lower ones. The keyer strips live on the **KEYS** page (ME1 has its own set on **ME1 KEYS**).

Fill source and mode

Each row in the **KEYERS (KEY1 bottom → KEY4 top)** group reads:

KEY n | FILL SOURCE | MODE | KEY CUT | KEY MIX | ON AIR | ADJUST

1. Click the **FILL SOURCE** dropdown and pick the source that carries the graphic: an NDI source, Test Pattern (internal), Still Store (internal), Media Player (internal), and so on. A key with no fill cannot go on air — its buttons stay inert.
2. Pick the **MODE**:
 - **Full** — the fill is used as-is at full opacity (a full-frame insert).
 - **Alpha** — keys on the fill's own alpha channel (PNG/still graphics, two-stream media with alpha).
 - **Luma** — keys on brightness; dial in with CLIP and GAIN in KEY ADJUST.
 - **Chroma** — keys on a chosen colour (green/blue screen work).
3. The **ON AIR** column tallies the key's live state.

KEY CUT and KEY MIX — the everyday controls

Each key has a direct pair, independent of the transition arm:

- **KEY MIX** (amber) — soft dissolve on/off at the transition RATE. The everyday control: bring lower thirds and bugs up and down on a MIX.
- **KEY CUT** (red) — instant snap on/off. Keep it for the moment you need it gone *now*.

Neither button arms or consumes the NTS. Pressing either again mid-fade snaps that key's move to completion. The same pair repeats in the transition cluster's **KEY TAKE** grid (next to the T-bar) for the master M/E.

Next-transition arming

To take a key with the background move instead:

1. Press **K2** (for example) in the transition cluster — the amber lamp lights next to **BKGD**.
2. Fire **CUT** or **AUTO** as usual: armed keys *toggle* (off→on, on→off) together with the background transition, sharing one timeline.
3. The arm is sticky and shows in look-ahead: with K2 armed, the PST monitor previews the key's *toggled* state so you can rehearse the take.

Keys always dissolve — WIPE/DME/STINGER apply to the background only.

Alpha interpretation — STRAIGHT or PREMUL

Inside KEY ADJUST's MATTE page, two interpretation buttons set how the fill's alpha is read:

- **STRAIGHT** — colour and coverage travel separately (typical PNG with alpha, most design-tool exports).
- **PREMUL** — colour is already multiplied by alpha (ProRes 4444 premultiplied exports, matted graphics).

Match the source or edges fringe: premultiplied content read as STRAIGHT darkens the edge; straight content read as PREMUL washes the edge out. The choice persists per keyer in sessions, macros and config; stinger imports auto-detect it.

KEY ADJUST — the per-key DVE

Press **ADJUST** on a keyer row to open the floating *Key Adjust — KEY n* window (pressing ADJUST on another key re-targets the same window). Edits show live on PST (and on PGM if the key is on air) and never arm or take the key. Four tabs:

- **MATTE** — by mode: ALPHA greys out (nothing to tune); LUMA gives CLIP and GAIN; CHROMA gives **KEY COLOUR** (with the ⊕ pick from PVW eyedropper — arm it, then click the colour on the PST monitor), **TOLERANCE**, **SOFTNESS** and **SPILL SUPP.**; plus the **STRAIGHT/PREMUL** pair and the **KEY INVERT** toggle. **MATTE VIEW** routes the selected key's alpha to the preview monitor as greyscale — the edge/holes judge while you dial a chroma key (preview only, never on PGM, not saved).
- **TRANSFORM** — **POSITION X**, **POSITION Y** (−1...+1 of frame) and a uniform **SIZE** (10–400 %) — reposition/scale a PiP or lower third.
- **CROP** — **TOP / BOTTOM / LEFT / RIGHT** insets toward the centre.
- **FRAME** — a prefab edge overlay scaled to the key's box: **FRAME PRESET** (None, thin_white, thick_white, rounded, bevel, drop_shadow) or **Load PNG...** for a custom full-canvas RGBA frame (transparent centre, opaque edge).

KEY ADJUST — KEY 2

MATTE

TRANSFORM

CROP

BORDER

MODE: CHROMA

KEY COLOUR

#123456

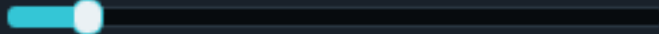
⊕ pick from PVW

TOLERANCE



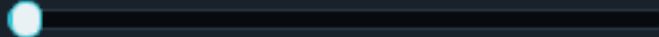
1.00

SOFTNESS



0.10

SPILL SUPP.



0.00

KEY INVERT

MATTE VIEW

edits show live on PVW - does not put the key on air

KEY ADJUST — KEY 2

MATTE

TRANSFORM

CROP

BORDER

POSITION X

0.00

POSITION Y

0.00

SIZE

100

edits show live on PWV - does not put the key on air

KEY ADJUST — KEY 2

MATTE

TRANSFORM

CROP

BORDER

MODE: CHROMA

KEY COLOUR

#00B140

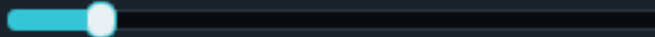
⊕ pick from PVW

TOLERANCE



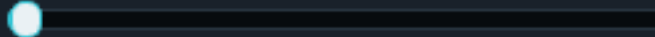
0.35

SOFTNESS



0.12

SPILL SUPP.

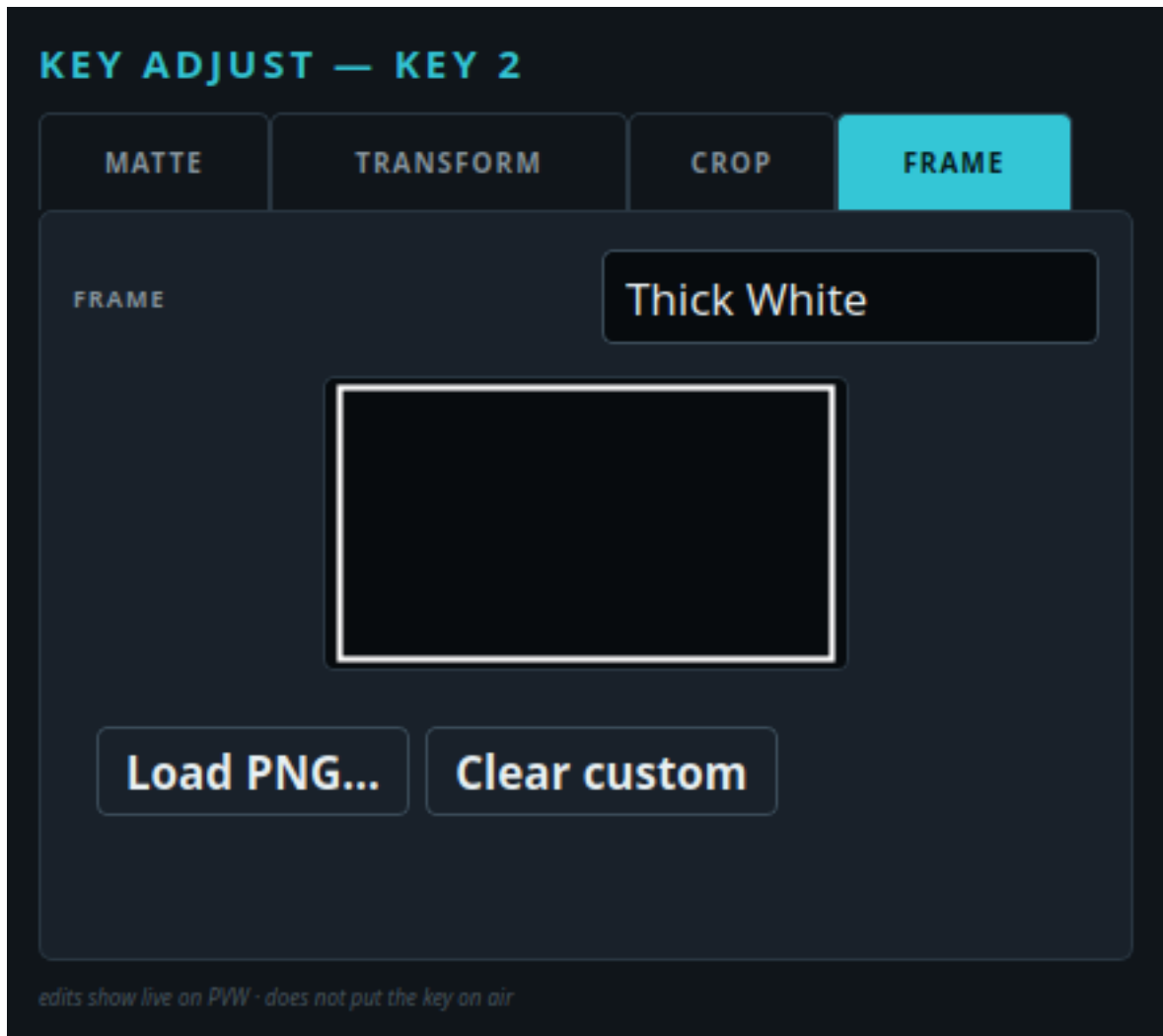


0.00

KEY INVERT

MATTE VIEW

edits show live on PVW - does not put the key on air



Known limit: chroma SPILL SUPPRESSION is weak on warm/red-dominant subjects (skin) — it pulls the key channel only where it dominates. It is a keyer algorithm limit, not a KEY ADJUST fault.

Setting up LUMA and CHROMA keys without a rig

The internal generator ships keyer-setup patterns (pick them as the fill):

1. Set the fill to Test Pattern (internal) and the **TEST PATTERN** dropdown to **Luma Ramp** — feeding it to a LUMA key makes CLIP/GAIN obvious: clip sets where along the ramp the key starts, gain sets the softness.
2. For CHROMA, pick **Chroma Green Field** or **Chroma Blue Field** — flat full-frame fields at the keyer's default target colours; confirm tolerance and edge produce full transparency on the field before pointing the key at live video.

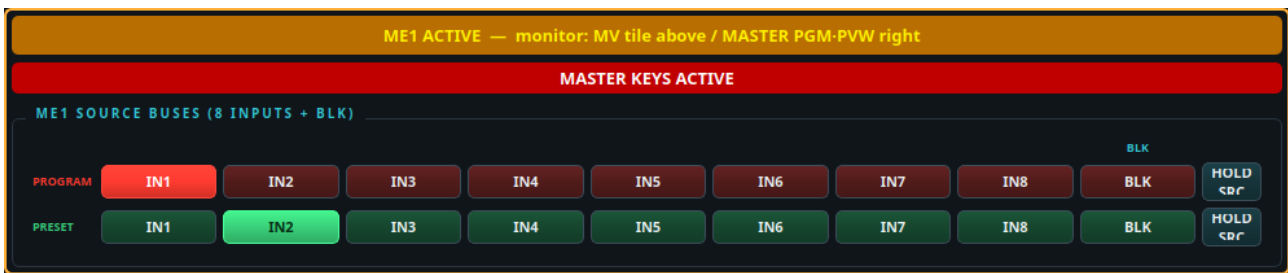
ME1 and E-MEM

The second M/E and the effect-memory system. **ME1** is a full second mix engine: its own buses, keyers and transition cluster, re-entered into the master as a source (the **ME1** column on the master crosspoints).

The amber ME1 page

1. Press **ME1** in the header navigation. The page is ringed in **amber** — an anti-confusion cue that you are now controlling the *second* M/E, not the master.
2. The amber **ME1 ACTIVE** pill confirms it, and tells you where to watch: the **ME1 tile** in the multiviewer (left column) and the master PST/PGM monitors when ME1 is routed there.
3. The page mirrors the master layout:
 - **ME1 SOURCE BUSES (8 INPUTS + BLK)** — red **PROGRAM** and green **PRESET** rows (IN1–IN8 plus BLK, no ME1 self-select).
 - **ME1 EFFECTS TRANSITION** — **BKGD**, **K1–K4**, **MIX/WIPE/DME**, **REV**, the T-bar, **KEY TAKE**, **CUT/AUTO** and its own **RATE** (no FTB on ME1 — fade the master instead).
 - **ME1 KEYS** page — the four ME1 keyer strips with the same **FILL SOURCE / MODE / KEY CUT / KEY MIX / ADJUST** layout.
4. To put ME1 on the master: press the **ME1** button in the master's **PROGRAM** row (instant) or **PRESET** row (staged), like any source.

MASTER KEYS ACTIVE — the red lamp



The red **MASTER KEYS ACTIVE** banner under the ME1 pill lights whenever any *master* keyer is composited over the ME1 re-entry. If an ME1 key "didn't come off" — you cut it but it still shows on the output — glance here first: it is almost always a master keyer riding on top of ME1, not an ME1 fault. Clear the master key (its **KEY CUT**) and the output cleans up.

The EFFECT BANK keypad

The **EFFECT BANK** block (right column, beside the transition cluster) is the E-MEM system: memory registers that recall a whole switcher look with one digit press. The display strip shows the current bank and membership, for example **BANK 0 · PGM**.

- **Digits 0–9** — register addresses. Ten registers per bank.
- **bank** — press bank, then a digit to *select a bank* (banks 0–9).
- **LEARN** — press **LEARN**, then a digit to *store* the current state into that register.
- A bare digit press — *recalls* that register instantly.
- **Me1 / PGM** (master page only) — membership: which M/E a store/recall acts on. Both lit = one button swaps both M/Es. PGM is lit by default.
- **AUD** — include the *audio fader scene* in the register: LEARN captures the whole mix (faders, mutes, pans, program master); a recall tweens the mix to it over the effect dissolve (or the audio recall time). With no M/E lit, an audio-only register recalls just the mix.
- **Eff** — *animate*: recall tweens continuous parameters (wipe positions and similar) over the transition instead of snapping.

- **AUTO RECALL** — latching mode where recalls follow the transition (master page only).

Store a look (the everyday E-MEM flow)

1. Build the look: buses, keys, transition type — anything you want recalled.
2. Press **LEARN** (red). The display shows LEARN armed.
3. Press a digit, for example 3. The state is stored into bank/register 3; the display confirms the action.
4. Press **LEARN** again or simply move on — the arm clears after the store.

Recall a look

1. Pick the membership (PGM, Me1, AUD as needed).
2. Press the digit of the register. The stored look lands: sources, keys, and (with **Eff** lit) animated transition parameters.
3. bank + a different digit first if the register lives in another bank.

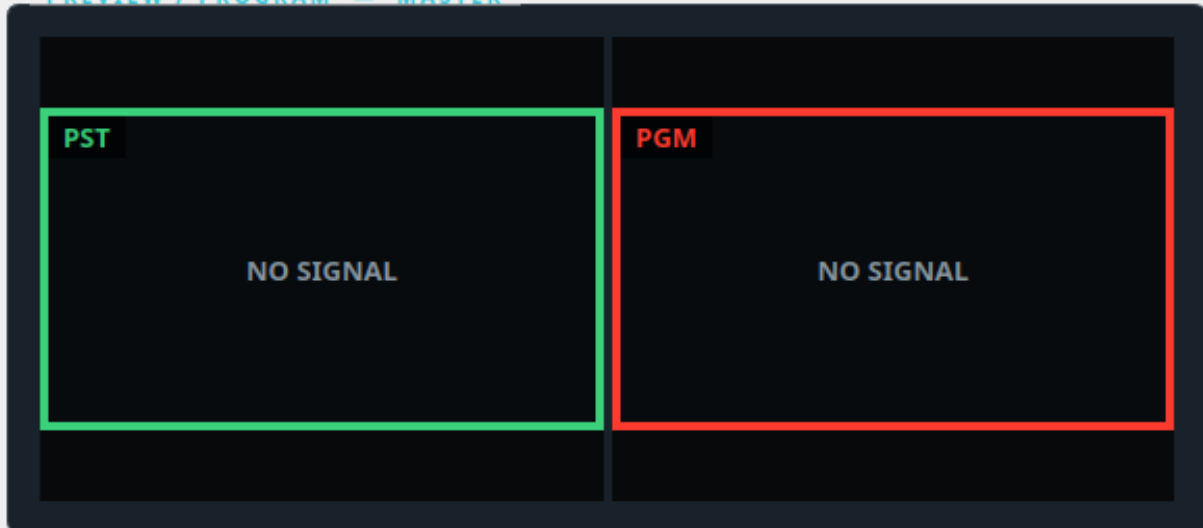
HOLD SRC — protect sources across recalls

Every bus and keyer has a **HOLD SRC** lamp (the PROGRAM/PRESET rows end with one, and the KEY TAKE grid has one per key). Lit, an E-MEM recall keeps that element's *source* as-is and recalls only the rest of the look — so a show register can change the look without re-routing a live camera. The lamps are latching and persist with the effect state.

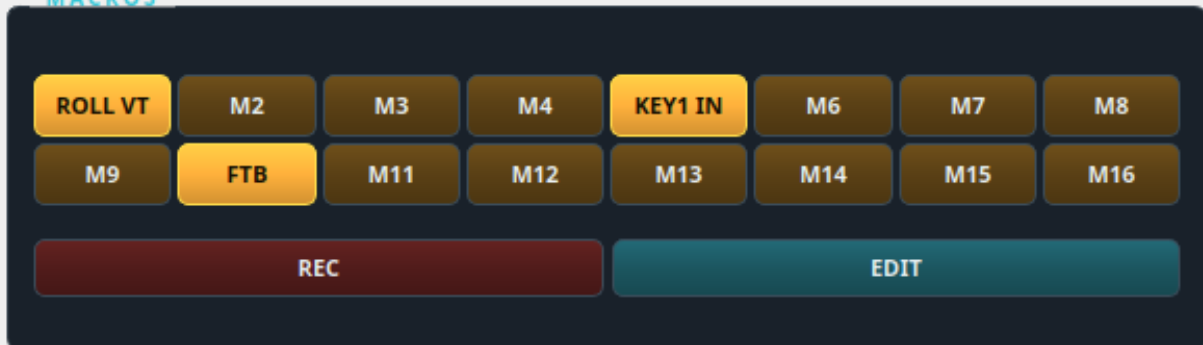
Macros

Sixteen macro slots on the right column, under the PST/PGM monitors, record and replay button sequences — the opening titles roll, the sting-plus-take, the end-of-show fade — at one press.

PREVIEW / PROGRAM — MASTER



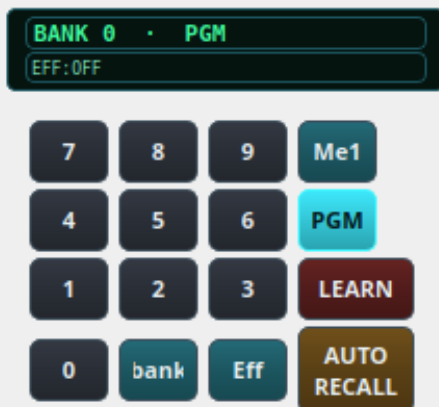
MACROS



EFFECTS TRANSITION



EFFECT BANK



The bank

- **M1 ... M16** — 16 slots in a 2×8 grid. A bare press **plays** the macro. Occupied slots light amber and show their name (e.g. ROLL VT).
- **REC** (red, latching) — arm recording.
- **EDIT** (cyan, latching) — open the editor.



Record a macro

1. Press **REC** — it lights red.
2. Press the slot to record into, for example **M1**. The slot lights red: the mixer now records your actions with their frame timing.
3. Operate the panel: takes, key cuts, still selections, AUTO moves, caption takes, and so on. Discrete commands are recorded — continuous gestures (T-bar scrubs, KEY ADJUST slider drags) are not.
4. Press **REC** again (or press the recording slot) to stop and store. The slot lights amber with its default name.

Replay a macro

Just press the slot button. The recorded commands replay with their recorded frame waits. A macro can also be bound to a keyboard key (below) and fired from the control socket (macro_play).

Edit a macro

1. Press **EDIT** (cyan).
2. Press the slot. The macro editor opens; **EDIT** un-arms itself after opening.

NAME

OPEN SEQ

wait (frames)		command
0 f	⬆⬇⬆	TAKE PGM slot 1
12 f	⬆⬇⬆	CUT (PGM)
0 f	⬆⬇⬆	KEY1 ON (PGM)
38 f	⬆⬇⬆	LOAD sting.mov
25 f	⬆⬇⬆	FADE TO BLACK

Add
Remove
Move Up
Move Down

OK
Cancel

In the editor:

- **NAME** — the slot's label (shown on the bank button).
- The event table — one row per recorded command with its **wait (frames)** spinner (the pause *before* that command fires).
- **Add / Remove / Move Up / Move Down** — edit and reorder rows.
- **Attach** — bind a keyboard key to the macro: click Attach, press the key. **Clear** removes it. Bound keys play the macro from anywhere on the panel (not while typing in a text field).
- **OK** stores the edited macro.

Conditions

A macro can be gated so it only fires when the mixer is in a given state:

- A **global condition** on the macro, and/or a **per-event condition** on any single row (built in the editor's condition column). A condition is a list of rules combined with AND; an empty list means "always".

- A rule reads a state path (e.g. a keyer's on-air flag, a bus source) and compares it; any rule can be inverted with *not*.
- At play time, a macro whose global condition fails does nothing; a row whose per-event condition fails is skipped, so a macro can behave differently depending on what is on air.

Fader memory (per-strip audio snapshots)

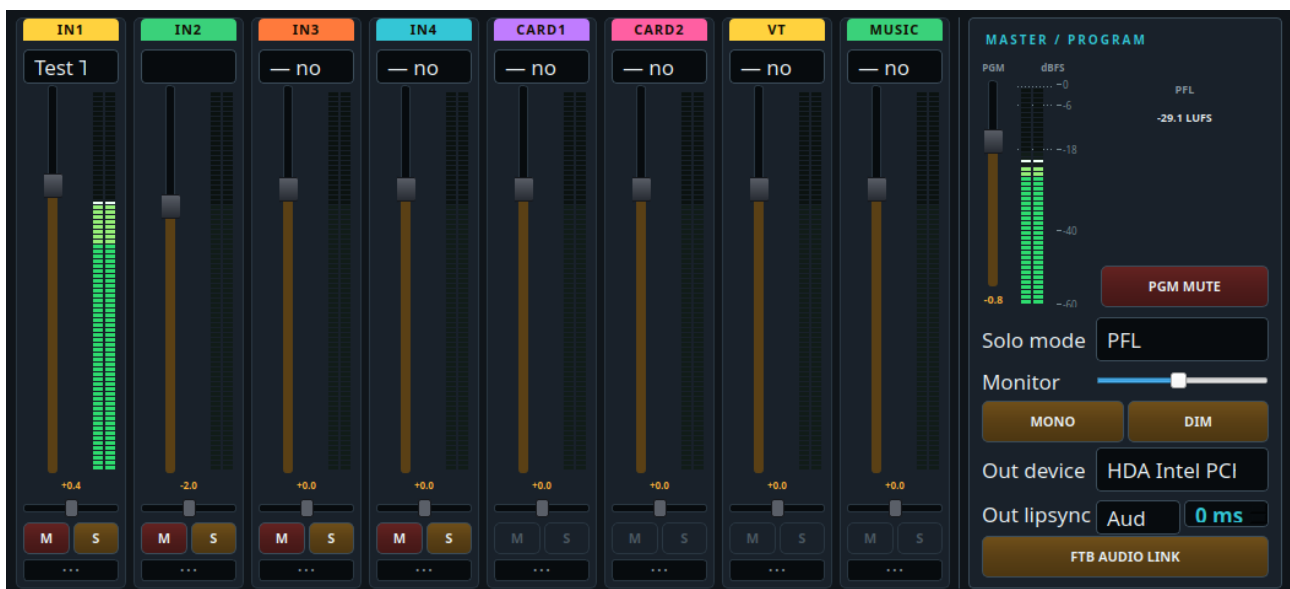
On the audio console, every assignable strip carries four memory lamps (1–4) and a red **L** (learn) key — a per-channel preset system, the strip-level counterpart to E-MEM scenes:

1. Build the strip's identity: bound source, fader, mute, pan, trim, EQ and dynamics.
2. Hold **L** about two seconds — it flashes red when armed (arming is one-shot and auto-cancels after a few seconds).
3. Press a memory lamp 1–4 — the snapshot is written to that slot (overwriting without asking; the green lamp is the cue) and saved to disk immediately.
4. Tap a lamp to recall: the strip's whole identity snaps back instantly (no ramp — it is a "bring back this exact channel" preset). An empty slot is dim and does nothing.

Fader memory, E-MEM audio scenes and macros are independent: an E-MEM recall does not touch learned strip slots, and a strip slot recall does not touch the E-MEM registers.

Audio

The audio mixer is a sixteen-strip broadcast console in a free-floating window. Open it with the amber **AUDIO MIXER** button (multiviewer strip header). Program audio is all-manual: what you fade up is what goes on air, independent of the video cut.



The sixteen strips

- **IN1...IN8** — the audio embedded in NDI video inputs 1–8. Each strip follows its video slot: re-point input 3 and channel IN3's audio follows.
- **CARD1 / CARD2** — external audio-card inputs (sounddevice/PortAudio).

- **VT / MUSIC** — assignable NDI audio-only sources (a comms mix, a playout feed).
- **MEDIA / APL / STG** — the internal sources, hard-wired to the media player, the audio player, and the stinger audio (no source picker — they carry audio the moment their feature plays).
- **TONE** — the internal 1 kHz alignment tone, starts muted.

Every strip carries, top to bottom: a source picker (where assignable), the  config flyout button, the fader with its meter, the memory lamps, and the **M/S** row.

Fader, mute, solo, pan

- **Fader** — dB scale, $-\infty \dots +10$ dB, unity at 0 dB. Default 0 dB.
- **M** (red) — mute. A muted channel still meters its input, and PFL still hears it (that is the point of PFL).
- **S** — solo to the monitor bus. A click latches; press-and-hold is momentary. Solo **never touches program** — it only changes what *you* hear.
- Solo mode is global: the monitor section's **PFL/AFL** selector. **PFL** (default) taps pre-fader, pre-mute, pre-pan, mono-summed — the "check this source" tool. **AFL** taps post-fader, post-pan stereo. A **PFL ACTIVE** tally makes it unmistakable that the monitor is on solo.
- **Pan/balance** — in the  flyout: -1 (hard L) to $+1$ (hard R), centre default.

EQ and dynamics

The **EQ/DYN** button on each source-carrying strip opens the channel's *EQ / Dynamics* window (it shows GR n while processing is reducing gain):

- **EQ** — HPF (off/20–300 Hz), low shelf, mid peak (with Q), high shelf, each ± 15 dB, with a master enable.
- **Dynamics** — a stereo-linked compressor (threshold, ratio, attack, release, knee, makeup) plus a per-strip limiter with its own ceiling.
- Defaults are bypass — an untouched strip is bit-identical to no DSP. The program sum is still guarded by the master -1 dBFS limiter regardless.

Lipsync

A/V alignment lives in two places (both in ms and frames):

- **Per-input** — the  flyout's lipsync spinner ($0 \dots +400$ ms) delays that channel's audio to match a source whose video lags (the common encoder case).
- **Output** — the monitor section's output lipsync (± 400 ms): delay audio (default mode) or a small PGM video delay, to trim residual program offset.

Metering — EBU

- Alignment: **-18 dBFS = 0 VU**; the meter scale marks -18 (align) and the -23 LUFS loudness reference; red zone above -1 dBFS.
- Per-channel meters are pre-fader (signal present and aligned, wherever the fader sits); the program meter shows the true on-air signal (post master, post limiter), L and R, with peak hold.
- Line up with the internal **Test Tone 1 kHz** at -18 dBFS: put it on a strip, raise the fader until the meter reads alignment, and the strip is calibrated.

The monitor section

What you hear, separate from air:

- **PFL/AFL** selector (see above), **PGM MUTE** (the program master mute), and the program master fader.
- **DIM** — a –20 dB monitor dip for talkback moments.
- **MONO** — sums L+R for mono-compatibility checks. Monitor-only; program is untouched.
- Monitor output device selector + level, and the **FTB AUDIO LINK** toggle (fades program audio with FTB when enabled; default off).



Test tone and program audio on PGM

- The **1 kHz / –18 dBFS test tone** is an internal source selectable on any strip — the line-up and meter-calibration source.
- The mixed **program bus** (with a –1 dBFS safety limiter) is embedded on the **NDI PGM output**, riding the same NDI source as the program video. The Preset output carries no audio.
- The media player's audio arrives on its own strip (pre-bound to a spare channel by default) with a player-side on/off gate upstream of the channel — see the Media chapter.

Media, Stinger, Caption and OGrf

The content tools along the bottom of the left column and the floating media window. The media player needs the GPU compositor — without one it is greyed ("Media player unavailable"), and the rest of the mixer keeps running.

The media player

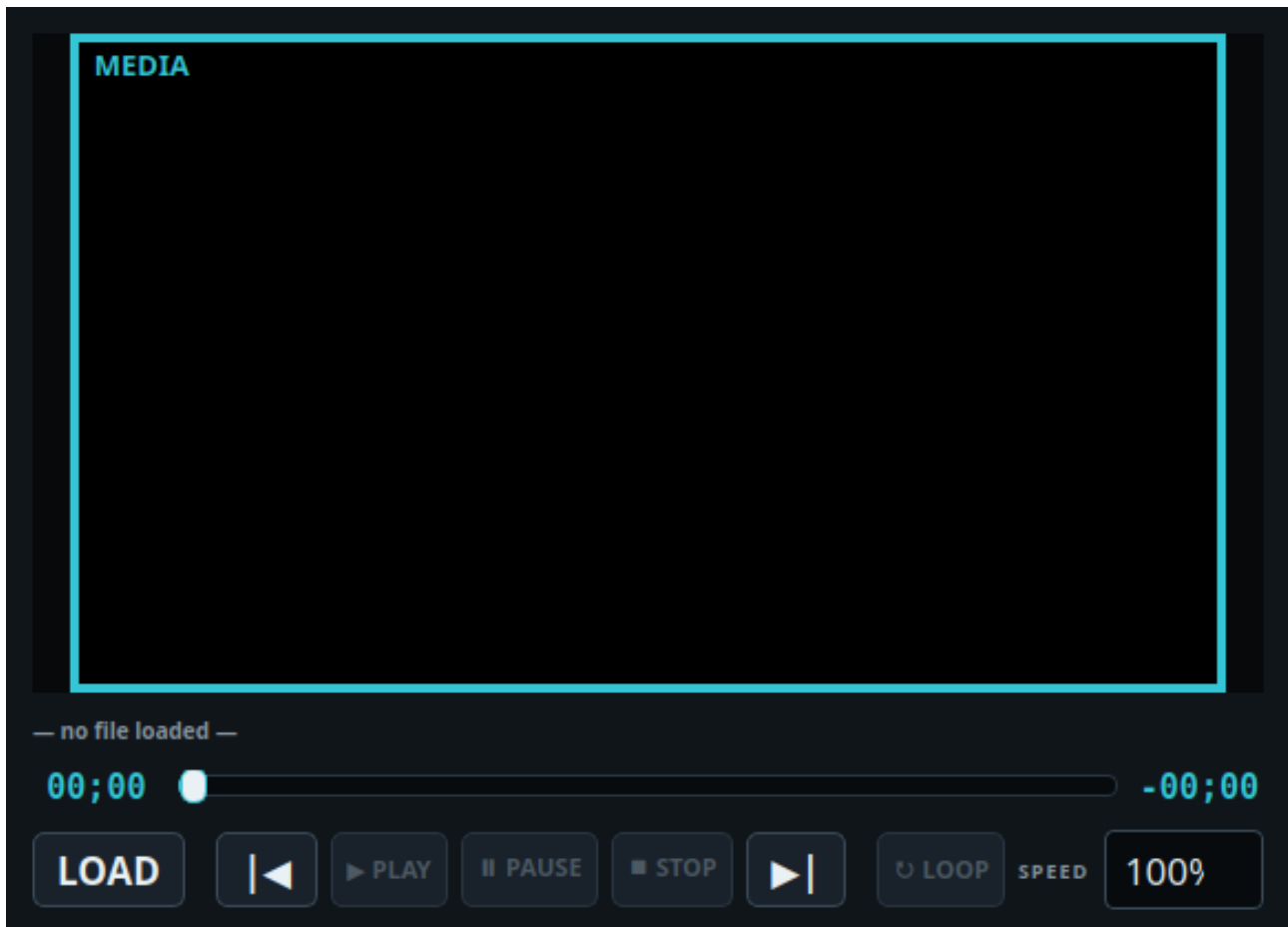
The **MEDIA PLAYER** group box is the quick transport; the **MEDIA** header button opens the full floating *Media Player* window with the playlist.

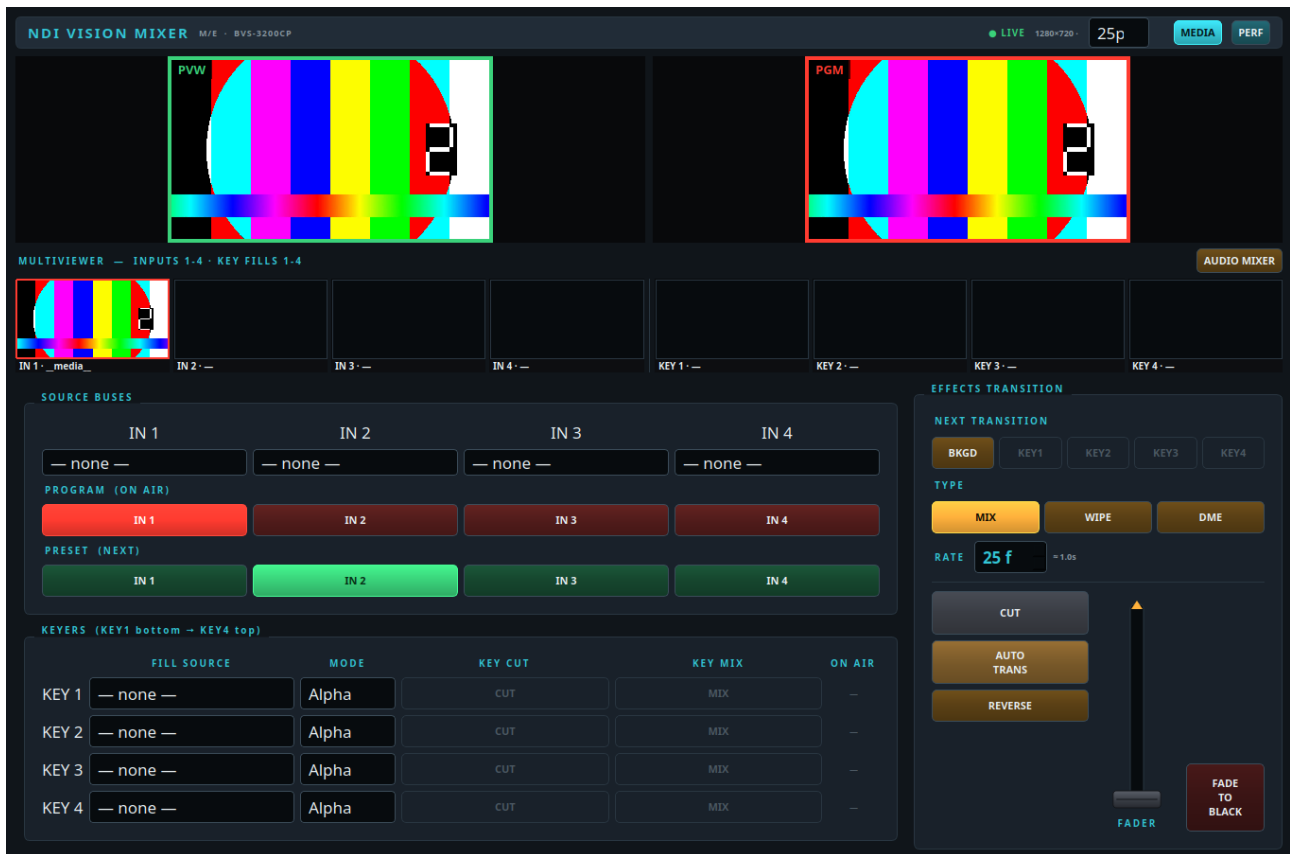
Quick transport (group box): **LOAD** (pick a clip), **■** (recue to first frame, paused), **■** play, **■** pause, **■** stop, **■** clip loop.

The floating window adds:

- The clip row with elapsed/remaining timecode and a scrub bar.
- **■ PLAY / ■ PAUSE / ■ STOP** and playback speed.

- ■ **LOOP** (clip), **AUTO-ADVANCE** (advance at clip OUT) and **PL-LIST** ■ (loop the whole playlist) — play modes off / clip / playlist.
- The playlist: add clips (+), per-clip **SET IN / SET OUT** trim, per-clip audio on/off, reorder — and **CLEAR**, which empties the playlist *and* parks the player (stop, eject, black) so nothing stays keyed on PGM.
- ■ **PGM** — take the media source to program in one press.





Notes worth knowing:

1. The **first import of a clip conforms it** (alpha two-stream $\approx 0.3\times$ realtime; opaque $\approx 5\times$ realtime). Later loads are cache hits.
2. Clip audio rides its own audio strip (a player-side gate + the channel controls in series). At speed $\neq 100\%$ the audio mutes.
3. At end-of-file (loop off) the player holds the last frame.
4. Off-output-fps files are frame-rate converted at conform (25p plays 1:1 into 50p as doubled frames — never $2\times$ speed).

Stinger transitions

A stinger is a pre-rendered animation (frame sequence + WAV) fired as a transition type:

1. Press **STINGER** in the type row (first press with nothing imported opens the import dialog).
2. **IMPORT...** — pick the clip; the alpha interpretation is auto-detected (colour under alpha 0 $\Rightarrow$ straight).
3. **CUT POINT...** — scrub to the frame where the hidden cut happens, and set it.
4. Arm **BKGD** and fire **AUTO** (or use the T-bar): the animation plays over the switch — the background changes under it at the cut point.
5. The stinger's WAV plays on a dedicated audio fader, fired with the take. Alpha interpretation can be overridden per stinger (straight/premul) in the import dialog.

Captions

The **CAPTION** group box typesets a one-line caption over a bar:

1. Type the text in the caption field.
2. **SET** — stages it into the preview well (proof it off-air; off-air edits are live as you type).
3. **TAKE** (red) — puts the caption on air as a keyed graphic.
4. Route it like any key fill: put the caption on a keyer (ALPHA mode is the self-key default) and **KEY MIX** it up, or take it with the arm.

OGraf graphics

GRAPHICS (OGraf) plays HTML-template graphics (lower thirds, animated score bugs) rendered by an OGrat subprocess:

1. Pick the graphic in the combo (or **■** to browse to a graphics library folder) and press **LOAD**.
2. The graphic's schema builds a live form — name fields, score spinners, clock, checkboxes. Type into it; the staged (edit) data is what **UPDATE** pushes.
3. **UPDATE** (red) — the on-air take: pushes the staged form data to air. It lights whenever the staged data differs from what is live (a hot pending-take cue). **PLAY** advances multi-step graphics (and is the step **■**); **STOP** clears the graphic.
4. Custom actions declared by the graphic appear as extra buttons (e.g. Flash, Set home score... — the ... ones open a small payload dialog).

The screenshot shows the 'GRAPHICS (OGraf)' control panel. At the top, there's a title bar. Below it, a text input field contains 'Test Caption (Lower Third)' next to a small icon and a 'LOAD' button. Underneath, the label 'Test Score Bug (animated)' is followed by several input fields: 'Home name' with 'ABC', 'Away name' with 'XYZ', 'Home score' with '2.00', 'Away score' with '1.00', and 'Clock' with '12:34'. At the bottom, there are three large buttons: 'PLAY' (teal), 'UPDATE' (red), and 'STOP' (teal). Below these are two smaller buttons with left and right arrow icons, and at the very bottom, two buttons labeled 'Flash' and 'Set home score.....'.

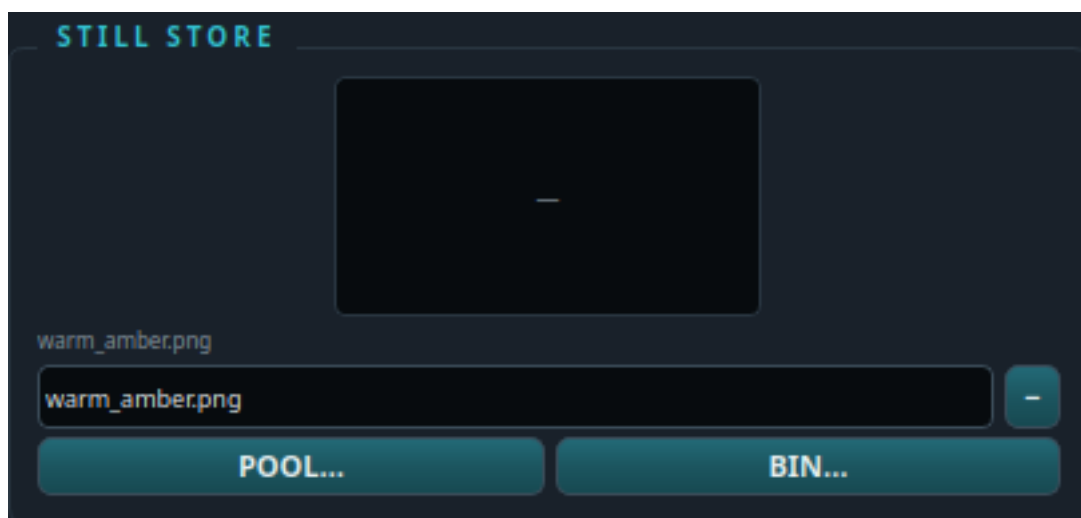


Launch note: the graphics renderer is enabled by default in the installed app. If it reports *renderer unavailable*, graphics publish transparent frames (the documented degrade mode) — restart the app to recover it.

Still Store

A bin of still images with **one** store output — `Still Store (internal)` — selectable on any bus or key fill like a source. Selecting a still is an instant hard-cut swap at the store; softness is an M/E transition (cue the `__still__` source on PST and take).

The pool

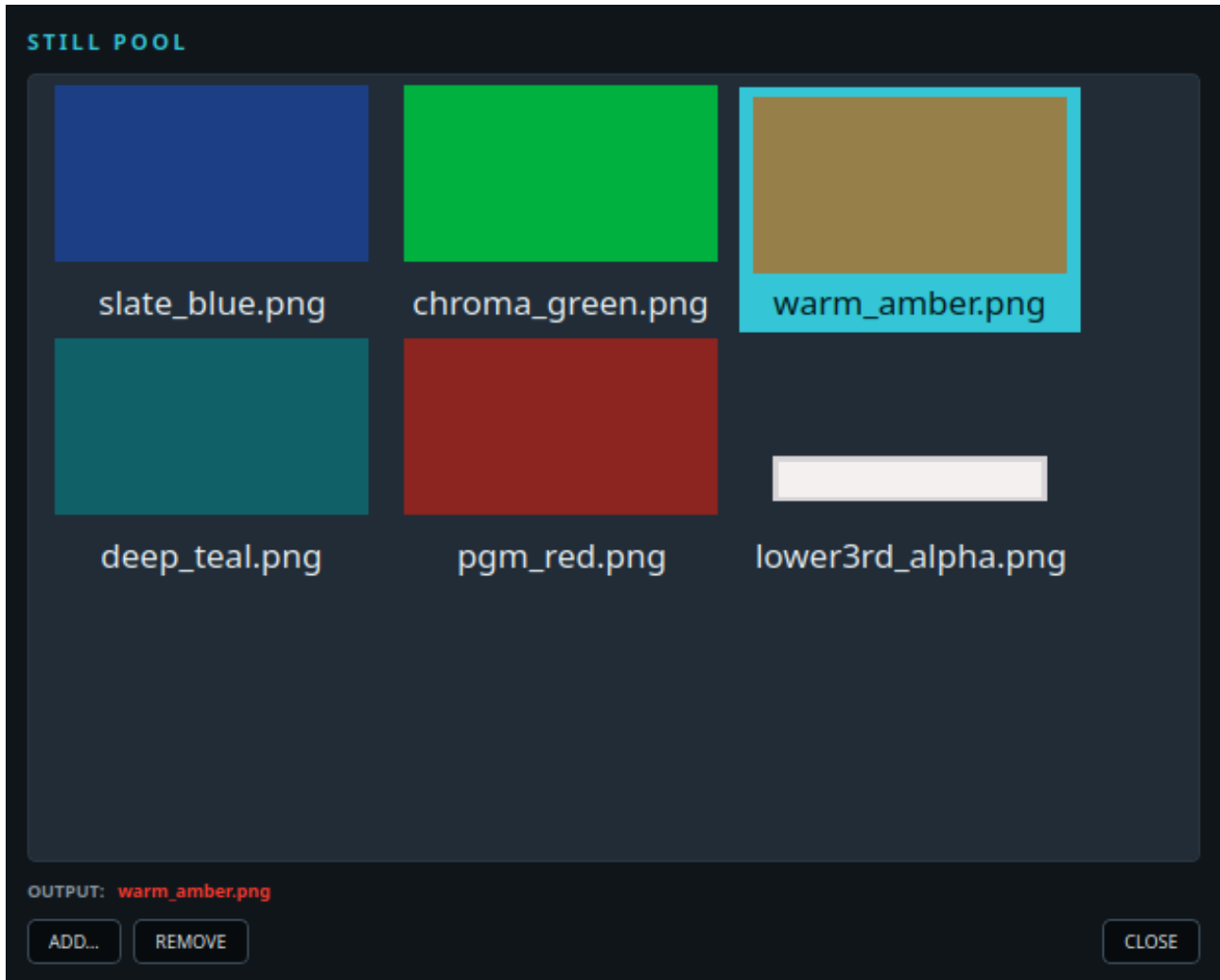


The **STILL STORE** group box shows:

- The preview well (the current store output) and the still's filename.

- The **pool combo** — every still in the bin. Picking an entry is the swap.
- — — removes the highlighted entry from the bin.
- **POOL...** — opens the thumbnail browser (below).
- **BIN...** — the classic file picker to load PNGs into the bin.

The POOL browser — click to swap



1. Press **POOL....** Every still renders as a 160×90 thumbnail tile.
2. **Click a tile** — that still becomes the store output immediately (the instant swap; the click is macro-recorded). The dialog **stays open**, so you can click through still after still against the monitors.
3. The highlighted tile and the red **OUTPUT:** row always name what the store is emitting.
4. **ADD...** appends files to the bin (it never clears it); the last one added becomes the output. **REMOVE** drops the highlighted entry. **CLOSE** when done.

The engine persists every add/remove/selection — the bin survives a restart and is captured by sessions.

Taking a still to air

The store is a *source*, not a bus — route it like any input:

1. As a **background**: press the `__still__` source's slot button on the **PRESET** row (after pointing that input's dropdown at `Still Store (internal)`), check it on PST, then **CUT/AUTO**.
2. As a **key**: set a keyer's **FILL SOURCE** to `Still Store (internal)` and **KEY MIX** it up — with **MODE Alpha**, the still's own transparency keys.

Formats and alpha keying

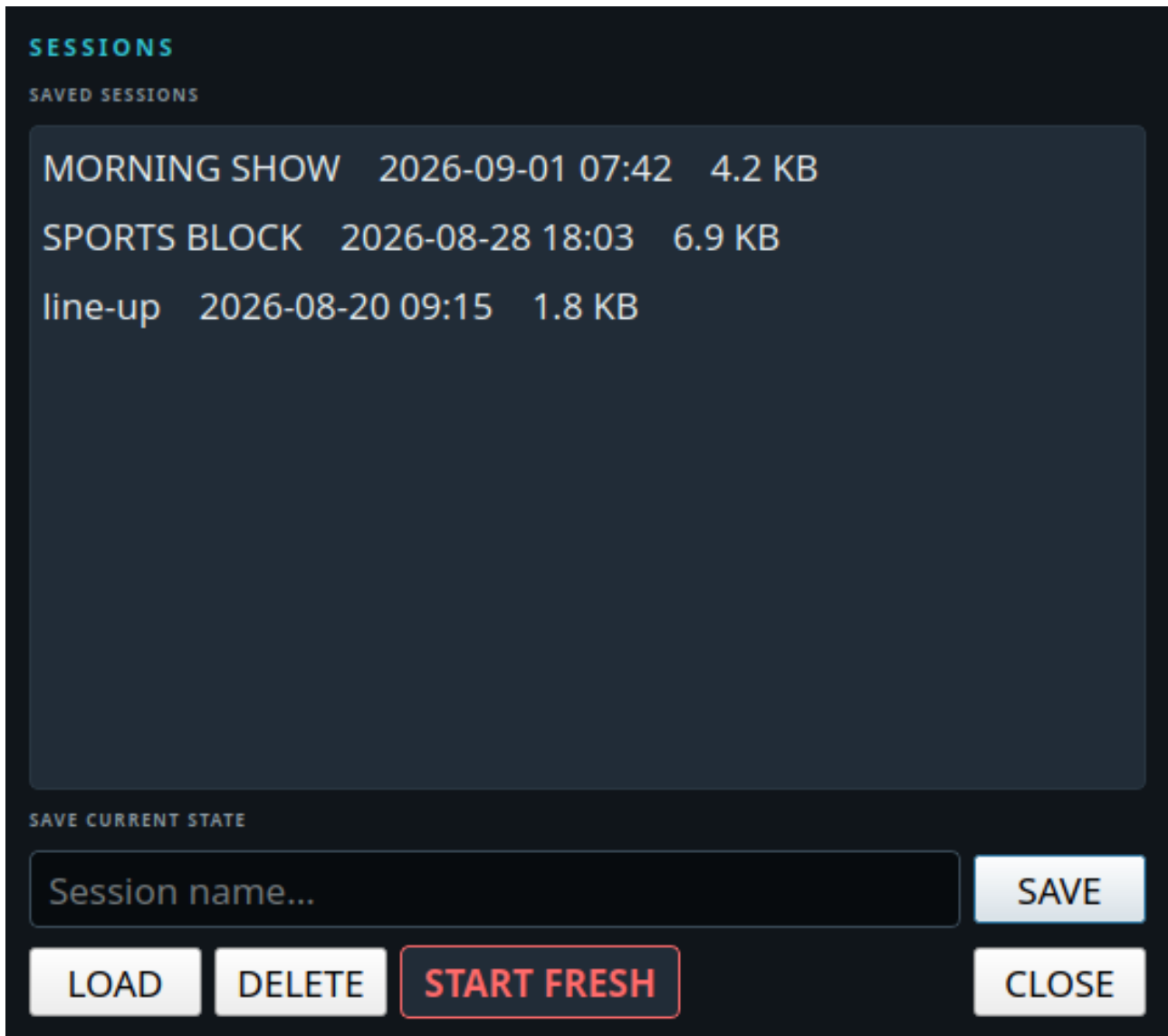
- **PNG with alpha** — the alpha channel keys (letterboxed into frame, aspect preserved, centred, transparent pad; the pad reads black over a background and keys out over a key). This is the lower-third/logo case.
- **PNG/JPG opaque** — a full-frame slate: use it as a background, or key it with LUMA/CHROMA modes like any video.
- Art is never cropped or stretched — a mismatched aspect letterboxes.
- An unloadable path renders as an opaque-black tile (visible, not a crash).

Persistence

The bin (paths + selection) is written to the mixer config on every change and restored on launch; sessions capture and restore it with everything else, and **START FRESH** clears it back to empty.

Sessions

Named snapshots of the whole mixer state — the new-show workflow is *START FRESH*, *build*, *save*; later shows *load* their session back.



Press **SESSIONS** in the header to open the dialog.

Save the current state

1. Build the show: buses, keys, sources, audio mix, E-MEM, macros, stills — everything you want captured.
2. Press **SESSIONS**.
3. Type a name in the field under **SAVE CURRENT STATE**.
4. Press **SAVE** (or press Enter in the field). The row appears in **SAVED SESSIONS** with its timestamp and size. Saving over an existing name asks for confirmation first.

Each session is a `.vmsession` file the engine persists; the dialog is a thin shell over the engine's session store.

Load a session

1. Click a session row to select it.
2. Press **LOAD** — or double-click the row.

3. The mixer restores to the saved state; the panel re-syncs immediately (the dialog closes on a successful load). If files the session references are missing (a still, a clip), a warning lists them — everything else still restores.

Delete a session

Select the row, press **DELETE**, confirm.

START FRESH — clean start from scratch

The red **START FRESH** button is the beginning-of-show reset. After confirmation it runs the engine's factory reset:

CLEARED (back to show defaults):

- buses, keys and input source assignments
- the audio mix
- E-MEM registers and macros
- audio fader memories
- media, stinger and FTB state

KEPT (machine setup, not show content):

- output format and frame rate
- NDI setup (including `extra_ips`)
- monitor device selection
- GPU keyer budget
- every **saved** session

Reset + load is the documented new-show flow: **START FRESH**, then **LOAD** the show's session (or build anew and **SAVE** it).

When sessions are your friend

- Rehearsal vs show day: save Line-up after building the routings; reload it any time the surface drifted.
- Multi-show boxes: one session per show (**MORNING SHOW**, **SPORTS BLOCK**); **START FRESH** between them keeps machine setup intact.
- Session files also drive the control socket (`session_save`, `session_load`, `session_reset`) — see the `vmctl` chapter.

Tools and Utilities

The helper surfaces: the internal test-signal generator, the router page, the multiviewer, the clean HDMI output, and the performance overlay.

Test patterns

The internal generator is an always-available source (no NDI gear needed), selectable on every input slot and key fill as `Test Pattern (internal)`. Pick the pattern with the **TEST PATTERN** dropdown

(visible when a slot uses the internal source):

- **EBU 75% Bars** (default) and **SMPTE bars** — line-up.
- **Black / 100% White / 75% Grey** — level and exposure reference.
- **Crosshatch** — geometry and aspect check.
- **Moving Box** — the motion/latency/dropped-frame check (a static pattern cannot show a stalled pipeline; a moving one shows it at a glance).
- **Chroma Green Field / Chroma Blue Field** — flat keyer-colour fields for CHROMA key setup.
- **Luma Ramp** — the LUMA keyer's setup pattern (clip/gain made visible).

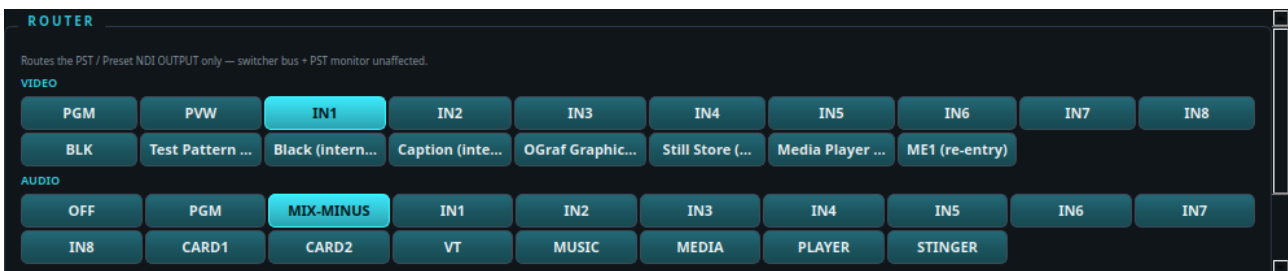
Generated patterns are fully opaque; ALPHA-keying them keys nothing, which is correct.

The ROUTER page

Press **ROUTER** in the header (or the **ROUTER** page tab). The router page routes the **PST / Preset NDI output only** — a clean aux feed for stage monitors, records or a guest feed:

- **VIDEO** row — one tally button per routable source; the lit button is routed.
- **AUDIO** row — the audio that rides the routed output, including *Mix-Minus*.
- **MIX-MINUS EXCLUDES** — one toggle per channel; lit (red) = excluded from the N-1 mix (so a return feed doesn't hear itself).

The router never touches the switcher bus or the on-screen PST monitor — switching and routing are independent.



Multiviewer

The left-column multiviewer tiles: **IN 1–8** (two rows), the **ME1** tile and the **M/E** (master program) tile. Each input tile tallies its state; the keyer fills are visible through the KEYS page previews. A tile showing **NO SIGNAL** means the source has no frame yet (or none assigned) — the mixer holds the last good frame on loss rather than dropping to black mid-show.

HDMI clean output

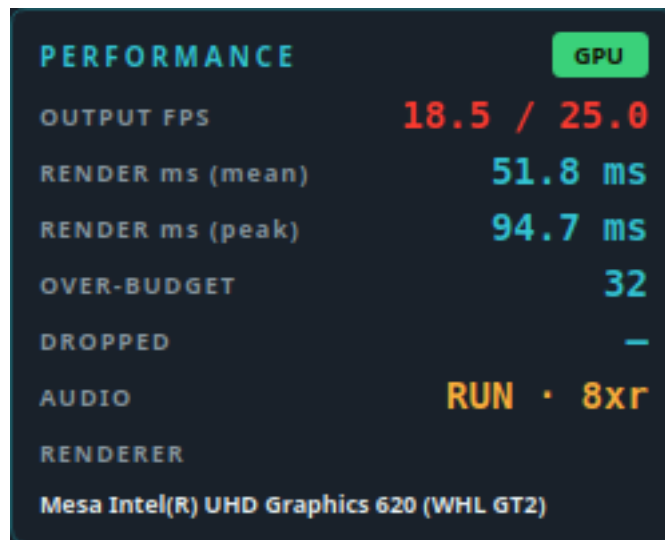
HDMI OUT ■ (header) opens a display picker:

1. Press **HDMI OUT** ■.
2. Pick the target display (e.g. **HDMI -1 — 1920×1080 (projector)**), or **OFF**.
3. The chosen screen shows full-screen **clean PGM** — the on-air composite with no UI, no tally, no overlays. The lamp stays lit while the output is live; the window is titled *NDI Vision Mixer — Clean PGM Out*.

PERF — the performance overlay

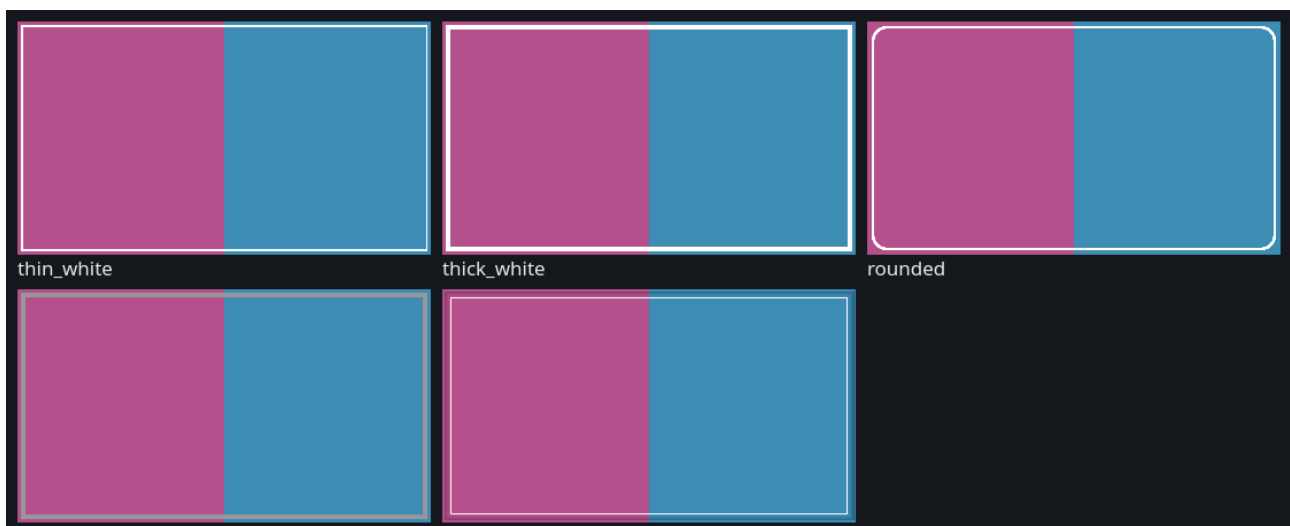
PERF (header) toggles a dismissible overlay with the render health at a glance: actual vs target fps ($\text{fps_actual} / \text{fps_target}$), render-time mean and peak, backend, and the send-fps truth. During a show the numbers to watch are:

- `fps_actual` at or above ~49.9 for a 50p show (the *50 fps law* — see Troubleshooting),
- render mean well under the per-frame budget (≈ 20 ms at 50p),
- zero underruns.



Frame presets preview

The KEY ADJUST FRAME tab's prefab overlays (thin/thick white, rounded, bevel, drop shadow) come from the shared frame registry — the same art renders on GPU and CPU compositors alike.



Remote Control (vmctl)

The mixer can be driven without GUI clicks over a localhost control socket — launching with `VM_CONTROL=1` opens it.

vmctl is **Linux-only**: the Windows build does not include the vmctl client.

```
VM_CONTROL=1 visionmixer
```

The control socket

- Path: /tmp/vmixer-control.sock (a local UNIX socket).
- Protocol: one JSON object per line, one JSON reply per line:

```
{"cmd": "auto", "args": [], "kwargs": {}}
```

- **Ints must be JSON ints.** A string "0" is rejected (and historically poisoned the render loop) — write 40, not "40". The same applies to booleans (true, not "true").

The vmctl client handles all of this:

```
vmctl <cmd> [args...]
```

Each positional argument is parsed as JSON if possible (so 40 → int, true → bool, STINGER stays a string) and passed through.

Common examples

Metrics / health (the 50 fps law numbers):

```
vmctl metrics
```

Sessions:

```
vmctl session_save "MORNING SHOW"
vmctl session_load "MORNING SHOW"
vmctl session_reset
```

Still store (index is an int):

```
vmctl set_still 2
vmctl still_load "/path/to/lower3rd.png"
```

Transition control:

```
vmctl toggle_nts_bkg
vmctl auto
vmctl cut
vmctl set_transition_rate 50
```

Stinger:

```
vmctl set_transition_type STINGER
vmctl stinger_set_cut_frame 40
```

Keys (per-M/E where supported):

```
vmctl set_key_mode 1 chroma
vmctl set_key_mode 1 chroma me=1
```

Reading state is the same shape — the reply is the JSON result the engine returns for the command.

Operational notes

- The socket is localhost-only; it exists while the mixer runs (a stale socket file after an unclean stop is removed on next launch).
- Every panel action has a command seam: if you can click it, vmctl can fire it — which is also how macros record and replay.

- The socket is **off by default**. Launch with `VM_CONTROL=1` to open it (`VM_CONTROL=0` keeps it off).

Installation and Operations

Setting Vision Mixer up and keeping it healthy.

Requirements

Vision Mixer runs on **Linux** (Ubuntu 24.04 or newer) and **Windows** (Windows 10 or 11, 64-bit).

Linux:

- **Ubuntu 24.04 or newer** (x86_64), with Python 3.12 or 3.13 (the standard system Python on 24.04+ — nothing to install).
- The **installer provides everything else**: all Python packages (bundled wheels), the NDI runtime (installed to `/usr/local/lib` and registered automatically), the audio (PortAudio) and media (FFmpeg) libraries. Windows:
- **Windows 10 or 11, 64-bit**.
- The Setup installer provides everything: the app, its libraries and the NDI runtime (installed next to the app). No separate NDI Tools install is needed.

Both:

- A GPU is recommended for smooth operation; the mixer calibrates itself to your hardware on first launch.

Frame rate and GPU class

Output rate	GPU requirement
25p (the calibrated default)	Runs solidly on mid-range workstation GPUs (e.g. laptop Quadro/RTX class), including media playback and keyers.
50p	Requires a desktop-class GPU. On mid-range mobile GPUs, 50p with media playback will not hold full frame rate — leave the output at 25p.

The mixer reports its own health live (`perf monitor / vmctl metrics`): `output_fps` should sit at your configured rate. Under extreme switching storms (hundreds of rapid cuts with media, keyers and loopback inputs combined) brief dips a few frames below rate can occur on mid-range GPUs — steady-state operation holds the line. This is the documented performance envelope for those GPUs.

Installing

Linux

```
sudo apt install ./visionmixer_<version>_amd64.deb # the file you downloaded
```

That is the whole setup — no internet needed at install time, no extra downloads, no NDI SDK step.

Windows

1. Download `Visionmixer-setup-<version>.exe` from **lumengrid.shop** (Vision Mixer page).
2. Run it. Windows may show SmartScreen or Defender warnings ("Windows protected your PC", unknown publisher, or that the app could be dangerous). That can happen with a new installer. Choose **More info**, then **Run anyway** (wording varies by Windows version).

3. Follow the Setup wizard. It installs to `C:\Program Files\Vision Mixer` by default, adds a **Vision Mixer** Start-menu entry and can add a desktop shortcut.

Starting

- Linux: from the **application menu**, click **Vision Mixer**, or run `visionmixer` in a terminal.
- Windows: from the **Start menu**, click **Vision Mixer**.

The first launch begins a **30-day full-feature trial**. When the trial ends (or any time before), enter your license key via the **LICENSE** button on the panel.

Configuration

- Config: `~/.config/vision-mixer/config.json` on Linux, `%APPDATA%\vision-mixer\config.json` on Windows — output size and `fps_n` (e.g. `1920×1080, 50`), the keyer budget, audio setup, the still bin, and NDI settings.
- `ndi.extra_ips` — list known sender IPs here so discovery queries them directly and never depends on mDNS:

```
"ndi": { "extra_ips": ["192.168.1.20", "192.168.1.21"] }
```

- Diagnostics log: `~/.cache/vision-mixer/logs/visionmixer.log` on Linux, `%LOCALAPPDATA%\Visionmixer\logs\visionmixer.log` on Windows.

Updating

Linux: download the new `.deb` and install it over the old one:

```
sudo apt install ./visionmixer_<new-version>_amd64.deb
```

Windows: download the new Setup `.exe` and run it; it installs over the old version.

Your config, sessions, still pool and license stay in your user profile — untouched by updates and safe across uninstalls:

- Linux: `~/.config/vision-mixer`, `~/.local/share/vision-mixer`
- Windows: `%APPDATA%\vision-mixer`, `%LOCALAPPDATA%\vision-mixer`

Uninstalling

- Linux: `sudo apt remove visionmixer` (closes a running mixer cleanly first).
- Windows: **Settings** → **Apps** → **Vision Mixer** → **Uninstall**. The uninstaller closes a running mixer first and removes only the program folder.

Licensing

Vision Mixer ships with a **30-day offline trial** from first launch — full features, no registration. To keep using it after the trial, enter a license key: **LICENSE** button (panel header) → paste the key → **Activate key**.

- Keys are issued by **Vision Mix LLC (Lumengrid)** after purchase and sent by email once the payment goes through. Plans: **\$19/month or \$149/year**.
- The licence is verified offline against the key embedded in the app — no network activation, no phone-home.

- Status is always visible in the LICENSE dialog (plan, days remaining).
- If the trial or a key has expired, the mixer shows the License dialog at launch and does not start until a valid key is entered. A session that opens licensed **never locks mid-show**.
- One active install per key. GPL components bundled with the app keep your GPL rights — see EULA.md and SOURCES.md in the installation directory.

Troubleshooting

Symptom-first fixes, then the health law and a glossary.

Symptom → fix

Symptom	Likely cause	Fix
No sources in the dropdowns / newly started senders don't appear	mDNS (Avahi) not running, or wedged after a sender was force-killed	Add the sender hosts to <code>ndi.extra_ips</code> in the config file (see <i>Configuration</i>) and restart the mixer; on Linux, check Avahi with <code>systemctl status avahi-daemon</code> and restart it with <code>sudo systemctl restart avahi-daemon</code>
Sources listed but won't connect; stale names	Same mDNS stale-record wedge	As above — shut senders down cleanly (SIGINT / close the app), not with a force-kill
Black output on PGM	FTB latched down, or BLK taken, or no frame yet	Check the FTB lamp (press FTB to fade up); check the PROGRAM row; a source with no frame yet shows NO SIGNAL and holds last-good
NO SIGNAL tile then picture returns	Source dropout	The mixer holds the last good frame on loss — no action needed; the tally tells you it dropped
Output fps below 49.9 at 50p, or underruns climbing	Render starvation (heavy keys, hot sources)	Reduce load: fewer active keyers / hot sources; check PERF overlay; see the fps law below
Audio lags behind picture by up to ~0.5 s under heavy load	Ingest rings fill under render starvation (bounded, self-recovering)	Reduce render load; <code>media_ring_ms</code> in metrics is the truth; it recovers when load clears
ME1 key "won't come off" (cut it, still on output)	A MASTER keyer composited over the ME1 re-entry	Look at the red MASTER KEYS ACTIVE lamp on the ME1 page; cut the master key
Picture holds ~180 ms after a seek	Decoder re-cue after the seek	Expected; only on operator-initiated seeks
25p clip plays at correct speed into 50p but every frame doubled	Conform re-timing (by design)	None — 25p→50p duplicates frames; 50p passes 1:1

Symptom	Likely cause	Fix
Stinger take with no clip imported	Nothing loaded	IMPORT... then CUT POINT... (the first STINGER click auto-opens import once)

The 50 fps law

The health bar for a 50p show:

- `output_fps` $\geq$ **49.9** sustained,
- **zero underruns**,

measured over a 15–30 minute soak (via `vmctl metrics` or the PERF overlay). `send_fps` alone can lie — repeated frames count as sends; the output fps is the truth. At 25p the budget doubles (≈ 40 ms/frame); 720p with 1–2 live keys and 1080p with a single key run at rate; heavier loads degrade gracefully by a few fps.

Glossary

- **PGM** — program: the on-air composite (NDI output `<machine>` (PGM)).
- **PST / Preset** — the staged-next bus; the PST monitor is a *look-ahead* preview of the post-transition picture.
- **NTS** — next-transition selection: the sticky BKGD/K1–K4 arm.
- **Flip-flop** — the bus swap at the end of a background transition.
- **FTB** — fade to black (latching master output fade).
- **T-bar** — the manual transition fader; the bowtie alternates its direction per completed throw.
- **KEY CUT / KEY MIX** — a keyer's direct instant / dissolved on-off pair.
- **ME1** — the second M/E, re-entered into the master as a source.
- **E-MEM / EFFECT BANK** — memory registers recalling whole switcher looks.
- **HOLD SRC** — recall-protect lamp: an E-MEM recall keeps that element's source.
- **Macro** — a recorded button sequence replayed at one press.
- **Fader memory** — the four per-strip snapshot slots on the audio console.
- **PFL / AFL** — pre-fade / after-fade listen: monitor-only solo.
- **Stills / the bin / the pool** — the still store's image set and its thumbnail browser.
- **Stinger** — an imported animation transition with a cut point and audio.
- **OGraf** — the HTML-template graphics subsystem.
- **vmctl** — the control-socket client (the `vmctl` command).
- **Conform** — the one-time transcode of an imported media clip into the mixer's internal format (cached).
- **Underrun** — an output frame that missed its deadline; the law says zero.