



Visibox Manual

Version 6.0.0

manual.spaceage.tv

Contents

VISIBOX MANUAL	4
Installing Visibox	6
Quick Start	8
Upgrading from Visibox 5	10
Visibox Terminology	13
The Visibox Interface	17
Working with Projects	25
Project Settings	29
Adding Songs	31
Setlists	35
Adding Clips	39
Clip Playback Options	47
Launch Modes	55
Poly Mode	58
Connecting Cameras & Video Inputs	61
Audio in Visibox	64
Output, Canvas, and Displays	71
Optimizing Visibox	77
Setting Up for Live Performance	80
Using MIDI in Visibox	84
Real-Time Control	89
Connecting an Elgato Stream Deck	95
Media Management & Optimization	102
Effects	106
Visualizers	116
Overlays	123
Lyrics	126
Settings	132
Audio Settings	135
Devices Settings	137
MIDI Settings	139
Output Settings	143
Canvas Settings	147
General Settings	154
Updates Settings	156
Troubleshooting	158
Support & Bug Reports	162

Your Account	166
Visibox Resources	171
AI Assistants	173
Command Line Interface	176
Visibox API	184
OSC	196
Lighting Consoles	202
Remote Pairing	207
Editors	210
Clip Editor Panel	212
Song Editor Panel	221
The Editor Window	226
Song Timeline	228
Visibox Menu Tour	234
Application Menu	235
File Menu	238
Edit Menu	240
Project Menu	244
Song Menu	247
Clip Menu	251
Play Menu	253
Window Menu	257
Help Menu	260



VISIBOX MANUAL

Welcome to Visibox!

Visibox is a video and audio playback system for live performance. It is the “instrument” you use to trigger visuals while you perform!

Use Visibox’s drag-and-drop interface to create a song list and assemble the video clips, images, audio clips, live camera feeds, HDMI video inputs, or generative visualizers to accompany each Song. While you perform, trigger your Clips with a MIDI device such as a drum pad, piano-style keyboard, or foot pedals, or use your computer’s keyboard, mouse, Stream Deck, or the API.

What’s new in 6.0

- **Poly Mode.** A Song can play several Clips at once. Triggering a Clip no longer stops the others, and six modes decide how the Clips share the frame: Blend, Mosaic, Stack, Horizontal Stripes, Vertical Stripes, and Spots.
- **Launch Modes.** Each Clip decides what pressing and releasing its trigger does. Trigger starts it and lets go, Toggle switches it on and off, and Gate holds it up only while you hold the control down.
- **Setlists.** Save more than one running order in a Project and switch between them, without dragging Songs around before every show.
- **Overlays.** Logos and lower thirds above everything the Output shows, placed on a nine-point grid with their own size, margin, and opacity. Assign one to a Project, a Song, or a single Clip.
- **Video walls.** Spread the Output across a grid of displays, each with its own scale, rotation, and flip. Lay the rig out visually in Settings and watch the screens follow as you drag.
- **NDI, DeckLink, Syphon, and Spout.** Send the Output over the network to an NDI receiver, out over SDI from a Blackmagic DeckLink card, or straight into OBS, Resolume, or TouchDesigner over Syphon or Spout.
- **Recording.** Capture the Output to a video file. Start it now, or arm it and let the next Clip you fire begin the take.
- **Real-Time Control.** Eight named Controls, each pointed at a Clip or Effect parameter and driven live from a knob, a fader, or an on-screen slider.

- **OSC.** OSC control over playback, navigation, and volume, with feedback you can subscribe to. Visibox announces itself on the network, so TouchOSC and Companion find it on their own.
- **Lighting Consoles.** Patch Visibox as a fixture on any desk that sends sACN or Art-Net, and cue Songs, Clips, intensity, volume, and hue from the console.
- **Song Timeline.** Write a Song's show as a script: cues that fire Clips, ride the volume, and send MIDI, all on the Song's own clock.
- **Lyrics.** Paste in a chord sheet in almost any format and Visibox reads it, renders it, scrolls it, and puts it on a screen the players can see.
- **In and out points.** Play a section of a video file instead of the whole thing, and loop cleanly inside it.
- **ISF Effects.** Shader Effects join the CSS ones, and the rebuilt Effects editor lets you tune an Effect on one Clip without changing it everywhere it is used.
- **Settings.** Rebuilt into nine categories, with the MIDI Map and remote pairing now categories inside it rather than windows of their own.
- **Stream Deck.** The Elgato plugin is the only way Visibox talks to a Stream Deck now. Built-in Stream Deck Mode is gone.

Free to use: Visibox has a free plan, and a 14-day trial that turns everything on. [Your Account](#) covers what each plan holds.

New to Visibox? [Terminology](#) defines every term this manual uses.

Visibox interface tour

[Watch this video: https://youtu.be/eKZ6XSVQsYs](https://youtu.be/eKZ6XSVQsYs)

VIDEO: An overview of Visibox: setup, interface, clips, MIDI, and more. [YouTube link](#)

For more about Visibox, visit: <https://spaceage.tv>

Installing Visibox

What you need

Mac	macOS 12 (Monterey) or later. Visibox is a universal app, so it runs natively on both Intel and Apple Silicon Macs.
Windows	Windows 10 or later, 64-bit.

There are no specific hardware requirements. Higher-resolution content and larger Projects benefit from a faster drive and a faster CPU and GPU. If playback is not smooth on the machine you have, [Optimizing Visibox](#) is the page to read.

Download

Go to [spaceage.tv](#) and use the download button for your platform. It always gives you the current release.

Older releases are on the [Legacy Downloads](#) page, which also carries the release notes for each one.

Install on a Mac

1. Open the downloaded disk image.
2. Drag **Visibox** into your **Applications** folder.
3. Eject the disk image and open Visibox from Applications.

Put it in Applications and leave it there. Visibox updates itself in place, and it cannot do that from a disk image, your Downloads folder, or anywhere else. Running it from the wrong place is one of the reasons an update check fails. See [Update Settings](#).

The first time you open it, macOS confirms that you want to open an application downloaded from the internet. Click **Open**.

Permissions macOS will ask for

Visibox asks for these when it first needs them, not all at once. Each one is refusable, and refusing costs you only the feature it belongs to.

- **Camera**, the first time you add a [Camera Clip](#).
- **Microphone**, the first time you use an audio input.
- **Screen recording**, if you turn on System Audio Input. See [Audio Settings](#).

Visibox also checks two System Settings that interfere with full-screen output, and offers to change them for you. Let it. See [Setting Up for Live Performance](#).

Install on Windows

1. Run the downloaded installer.
2. Visibox installs and opens on its own.

Windows may show a **Windows protected your PC** screen the first time. Click **More info**, then **Run anyway**.

The installer puts Visibox where it can update itself. If you unpack a copy by hand instead, updates will not work, and Update Settings tells you so. See [Update Settings](#).

Antivirus

Real-time antivirus reads every file Visibox writes, which includes large video files as they import and a fresh backup every five minutes. That slows things down, sometimes badly.

The first time you open a Project, Visibox notices which scanner is running and offers to exclude the Project folder. With Windows Defender it can do it for you, and Windows asks you to approve it. See [Antivirus and file indexing](#).

First launch

Visibox opens with no Project. Two good next moves:

- **Choose Help > Download/Open Demo Project...** and look around a finished Project. This is the fastest way in.
- **Follow the Quick Start**, which goes from here to a Clip on a screen in eight steps.

Signing in

Visibox has a free plan, and a 14-day trial that turns everything on. You do not need to pay to start, and you do not need to decide now.

Sign in from the Registration window, which Visibox shows you when it needs to. [Your Account](#) covers what each plan includes and how activation works across computers.

Visibox checks in with our servers from time to time to confirm your plan. It keeps working when the machine is offline, so a venue with no internet is not a problem. See [Working offline](#).

Updating

Visibox checks for updates on its own and asks before it downloads anything. It never interrupts a show: while the Output, Controller or Lyrics window is full screen, it does not check, does not open the update window, and does not ask you to relaunch.

Everything about updates, including how to check on demand and how to skip a version, is in [Update Settings](#).

Quick Start

Eight steps from a fresh install to a Clip on a screen. Each one links to the page that covers it properly, so read those later and get something running first.

If Visibox is not on your computer yet, start with [Installing Visibox](#).

1. Open the Demo Project

Choose **Help > Download/Open Demo Project....** Visibox downloads a complete Project with Songs, Clips, audio, Effects, and Visualizers already in it.

Do this before you build anything of your own. It is much easier to see what a Project is for when there is one in front of you. See [Resources](#).

2. Find your way around

The **Controller** is the window you perform from. Songs are the rows. The Clips in the selected Song are the tiles in the **Preview** at the bottom.

Take two minutes with [The Visibox Interface](#) before going further.

3. Get the Output onto a screen

The **Output** window is what the audience sees. Drag it to your second screen, then press  to put it full screen.

If you cannot find it, choose **Window > Output Window** (). See [Output, Canvas, and Displays](#).

4. Trigger a Clip

Click a Song to select it, then press . The first Clip in that Song starts playing, and you see it on the Output.

Keys  through  trigger the first nine Clips of the selected Song. The arrow keys move between Songs and Clips. The space bar stops. See [the Play menu](#).

5. Make a Song of your own

Choose **Project > New Song**, or press . Double-click its title to rename it.

Songs are how a Project is organized. One per song in your set is the usual arrangement, but a Song is just a group of Clips you want together. See [Adding Songs](#).

6. Add Clips to it

Drag video or image files from your desktop straight onto the Song row. Each one becomes a Clip.

You can also right-click the Song for a live camera, a Visualizer, a solid color, or an audio file. See [Adding Clips](#).

7. Attach a backing track

With the Song selected, press  and pick an audio file. It plays under whatever Clip is running, and the Song row shows its waveform.

Attach the track you are playing along to. For stingers and sound effects you fire by hand, use an [Audio Clip](#) instead. See [Attaching audio to a Song](#).

8. Play it from a controller

The number keys work, but you will want your hands free. Connect a MIDI controller, open **Settings > MIDI**, and map its pads to Clips.

A foot pedal, or anything else that sends MIDI, works the same way. A Stream Deck talks to Visibox through its own plugin instead. See [MIDI](#) and [Stream Deck](#).

Where to go next

- **Terminology** defines every term this manual uses. It is short and it saves time.
- **Launch Modes** decides what pressing and releasing a trigger does.
- **Poly Mode** puts several Clips on screen at once.
- **Setting Up for Live Performance** is the show-day routine.

Upgrading from Visibox 5

Your Projects open unchanged. Nothing needs converting and nothing needs re-doing.

What changed is the app around them. This page is for finding the thing you cannot find: what moved, what is called something else now, and what is gone.

For what is new rather than different, see [What's New in 6.0](#) on the home page.

Things that are called something else

Four Clip and Song settings were sentence fragments that only made sense once you read the value beside them. They are now noun phrases that name the thing they set.

Visibox 5	Visibox 6	Where
When Clip Ends	End Action	Clip
When Retriggered	Retrigger Action	Clip
When Resuming	Resume Action	Clip
When Audio Ends	Audio End Action	Song

The values inside them did not change. Loop is still Loop, Restart is still Restart.

Three more, elsewhere:

Visibox 5	Visibox 6
Placeholder Clip	Color Clip
Video Wall Stretch	Video Wall Controller Compensation
Background Options	Edit Background... , which now opens the same editor panel every other Clip uses

Things that moved

The menu bar caught up with the right-click menus

In Visibox 5 the Song menu in the menu bar showed far less than the same menu you got by right-clicking, so whole capabilities were reachable only by right-clicking. That is fixed.

The **Song** menu now also carries Add Audio Clip..., Add Camera Clip, Add Visualizer Clip, Add Color Clip, Mute Audio, Audio End Action, Poly Mode and Program Identifier.

The **Clip** menu takes the opposite approach. It handles the Clip itself, and each of its playback settings is set in one place: the panel behind **Edit Clip...**

Edit Song... and **Edit Clip...** now lead their menus instead of sitting in the middle. They are still the way to everything a menu cannot draw: the sliders, the color picker, and the Filters and Effects editors.

Settings were rebuilt

Settings is now nine categories in one window. The **MIDI Map** and **Remote Pairing** are categories inside it rather than windows of their own. See [Settings](#).

Output Window Size no longer sets the resolution

In Visibox 5, resizing the Output window resized the picture Visibox rendered, which meant a small window quietly gave a recording or a network output fewer pixels than it asked for.

Now the window size is just the window size. A windowed Output renders at a fixed size and the picture is fitted into whatever window you chose, with the overhang cropped. Full screen is unchanged.

If you want to see the whole canvas in a window that is not the same shape, turn on [Letterbox Output window](#).

Things that are gone

Built-in Stream Deck Mode. The [Elgato plugin](#) is the only way Visibox talks to a Stream Deck now. It does more than the built-in mode did, and Visibox ships with a copy so you install it from inside the app.

Worth knowing on the first day

Six changes that will not stop you, but will surprise you.

- **Dropping a single audio file on a Song now asks a question.** In Visibox 5 it always became the Song's backing track. It can now also become an [Audio Clip](#), so Visibox asks which you meant. Dropping several at once still goes straight to a backing track.
- **A Clip no longer stops the others by default.** That is [Poly Mode](#), and a Song set to Mono behaves the way Visibox 5 did.
- **Every Clip now has a Launch Mode.** Trigger is the Visibox 5 behavior and the default, so nothing changes until you ask for it.
- **Clips fire on the way down.** Visibox 5 waited for you to release the mouse. Visibox 6 fires the Clip the moment you press it, which is faster under the hand and is what lets you hold a Gate Clip open from the Clip row. If you prefer the old feel, turn off **Trigger Clips on Press** in [Settings > General](#).
- **The Windows Plugins folder moved.** Your `.vbfx` and `.vbvs` files now live in `%APPDATA%\Visibox\Plugins\Plugins\` instead of `%LOCALAPPDATA%\Visibox\Plugins\`. The old folder sat inside the install and was wiped when Visibox was uninstalled. Visibox moves what is there the first time 6.0 starts, so there is nothing for you to do unless you had a shortcut pointing at the old path.
- **Dragging a Clip needs its grip now.** Because the tile fires on press, dragging from the face of a Clip would play it. Hover a Clip in the Song list and a grip appears along its top edge. Drag from there to move the Clip, and drag Songs by the matching grip on their rows. Clips in the Preview Pane cannot be dragged at all while this is on.

Opening a 6.0 Project in Visibox 5

Do not, if you can help it. Visibox 5 does not know about Setlists, Overlays, Poly Mode, Launch Modes, Audio Clips or anything else 6.0 added, and saving from there loses them.

Visibox 5 will not warn you, and nothing on screen afterwards tells you what it dropped. If you need to run both versions on the same show, work on a copy of the Project.

Visibox Terminology

A Visibox Project holds Songs, and each Song holds Clips. Nearly everything else in Visibox is a setting on one of those three.

This page defines every term the rest of the manual uses, and points at the page that covers it in full.

Project

The Project is the top level. It holds your Songs and their order, the Clips inside them, and the settings that apply to the whole show. A Project is one file, so you can carry a show from one computer to another. See [Working with Projects](#).

Song

A Song is the container for a group of Clips. It usually stands for a piece of music, though it can be any unit of a performance: a scene, a segment, a talk.

A Song does not need an audio track. It holds the visuals that play while you perform, and you attach audio only if you want Visibox to play a backing track. See [Adding Songs](#).

Clip

A Clip is what you trigger, and usually what the audience sees. Clip is the general term, and every Clip is one of six types: a [Video Clip](#), an [Image Clip](#), a [Camera Clip](#), a [Color Clip](#), a [Visualizer](#), or an [Audio Clip](#). The type decides what the Clip plays. Everything else about a Clip, from its Launch Mode to its Effects, works the same whichever type it is. See [Adding Clips](#).

Audio Clip

An Audio Clip plays a sound file. It is triggered like any other Clip, so it is how you fire a stinger, a sound effect, or a second track on cue. That is different from the one backing track a Song can have attached. It plays black unless you give it a color or pair it with a [Visualizer](#). See [Audio Clips](#).

Camera Clip

A Camera Clip shows a live video feed from a connected input: a USB webcam, an HDMI capture adapter, or a virtual camera. If your computer sees it as a camera, Visibox can play it as a Clip. See [Cameras and Video Inputs](#).

Color Clip

A Color Clip is a solid color, most often black. Visibox has to be playing something to play a Song's attached audio, so a black Color Clip is how you play "nothing" during a track that needs no picture. An [Audio Clip](#) is the other way to have sound with nothing on screen. Color Clips also make good canvases for [Effects](#).

Image Clip

An Image Clip is a still picture, added the same way you add a video: drag the file in. Animated GIFs are the exception. Visibox converts them on import, so a GIF arrives as a [Video Clip](#). [Working with Projects](#) lists the formats Visibox reads.

Video Clip

A Video Clip plays a video file. It is the type most people mean when they say “clip”, which is why this manual writes “Clip” for the general idea and “Video Clip” when it means this one in particular. If the file carries audio, that audio plays unless you mute it, and In and Out points trim the Clip to the section you want. [Working with Projects](#) lists the formats Visibox reads.

Effect

An Effect is a real-time visual transformation applied to a Clip: motion, color shifts, glitch, distortion, or animation that reacts to the music. Effects never touch your original media files. See [Effects](#).

Visualizer

A Visualizer is a Clip that generates live, audio-reactive visuals instead of playing a file. Visibox has two engines, MilkDrop and ISF (Interactive Shader Format). See [Visualizers](#).

Playing several Clips at once

Several Clips playing at once is **polyoptic** playback, and **Polyopsis** is the ceiling on how many. The words come from synthesizers: a synth that plays one note at a time is monophonic, and one that plays several at once is polyphonic, from the Greek *poly*, many, and *phone*, sound. Visibox plays pictures rather than notes, so it needs the other Greek word: *opsis*, sight.

Term	What it means
polyoptic playback	Several Clips playing at the same time in one Song. Triggering one Clip no longer stops the others.
Poly Mode	The per-Song setting that decides how simultaneous Clips share the stage.
Mono	Poly Mode’s one-Clip-at-a-time value, and the default. A new Clip replaces the one playing.
Blend, Mosaic, Stack, Horizontal Stripes, Vertical Stripes, Spots	The six Poly Modes that let Clips co-play, each laying them out differently.
Polyopsis	The ceiling on how many Clips can play at once, from 2 to 8. One setting for the whole app.

Starting and stopping a Clip

Term	What it means
Launch Mode	The per-Clip setting for what pressing and releasing its trigger does.
Trigger	A Launch Mode: press starts the Clip, and the release does nothing.
Gate	A Launch Mode: the Clip plays only while you hold the control down.
Toggle	A Launch Mode: press starts the Clip, and a second press stops it.
End Action	What happens when a Clip's media runs out. Loop is the default; Stop (One-Shot) is the value that ends playback.
Retrigger Action	What a second press does to a Clip that is already playing.
Resume Action	Where a Clip picks up when you come back to it, including Sync to Song Position .

The three settings above are covered on [Clip Playback Options](#).

Shaping a Clip

Term	What it means
Fit Mode	How media fills the frame. Its values are Crop, Letterbox, and Stretch.
In Point / Out Point	The trim handles on a Video Clip or an Audio Clip. Set them to play a section of a file instead of all of it.
Loop Crossfade	A slider that blends the end of a loop into its start.
Transition In / Out	The shape a Clip uses to arrive and leave: Fade, Wipe, or Push.
Velocity Sensitivity	Maps how hard you hit a MIDI pad to how loud the Clip's audio plays.
Tweak	The panel holding the Effect parameters one Clip has changed, so a shared Effect preset stops changing everywhere at once.

Everything in this table lives on [Clip Playback Options](#), except Tweak.

Organizing a show

Term	What it means
Setlist	A saved running order of the Songs in a Project. A Project can hold several and switch between them.

Term	What it means
Overlay	A logo or lower third that sits above everything the Output shows. Assign one to a Project, a Song, or a single Clip.
Song Timeline	A timed list of cues attached to a Song, written as a script.
Timeline Editor	The window you write a Song Timeline in.
Lyrics	The chord-and-lyric sheet feature, shown in the Lyrics Window on its own screen.
Project Settings	The window holding the settings that belong to one Project: its title, Uppercase Titles, Default Fit Mode, and its Controls.

Controlling Visibox live

Term	What it means
Real-Time Control	Named Controls pointed at Clip and Effect parameters, so a knob or fader can ride them during a show.
Control	One slot in the Control Bus. Name it, point it at a parameter, and drive it from MIDI, OSC, or an on-screen fader.
Control Bus	The set of Controls a Project has, and the Project Settings tab where you name and shape them. Every Project starts with eight, lettered A to H, and you can add more.

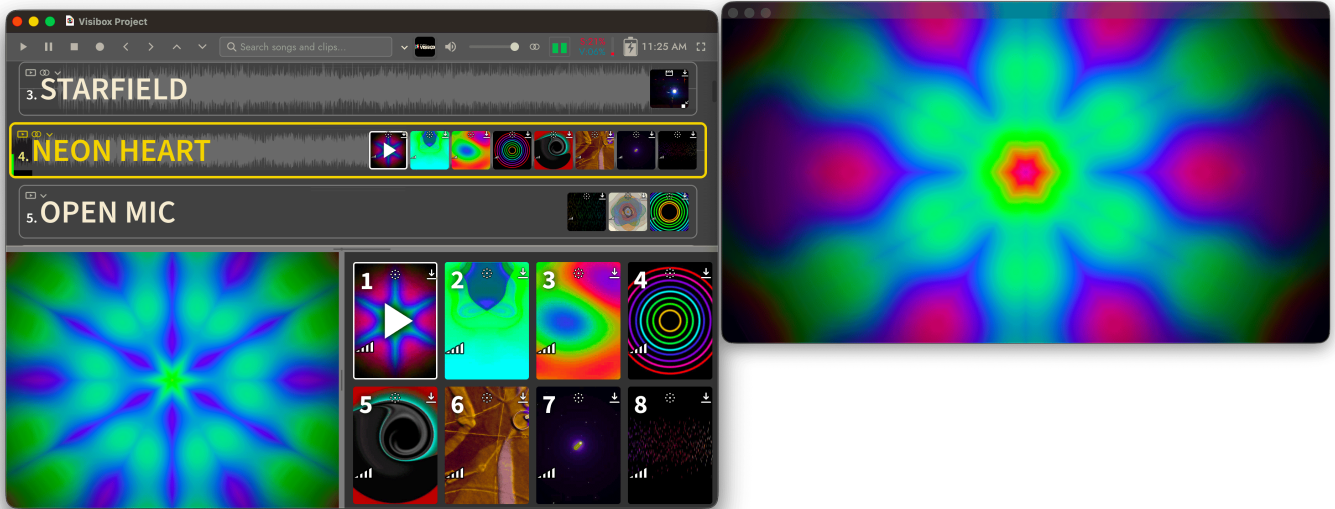
The Output

Term	What it means
video wall	The Output spread across several displays arranged in rows and columns.
Video Wall Controller Compensation	Pre-distortion for a hardware wall controller, set per display in Canvas Settings .
Record	Capturing what the Output shows to a video file.

Next, take a look around [the Visibox interface](#) before moving on to the detail.

The Visibox Interface

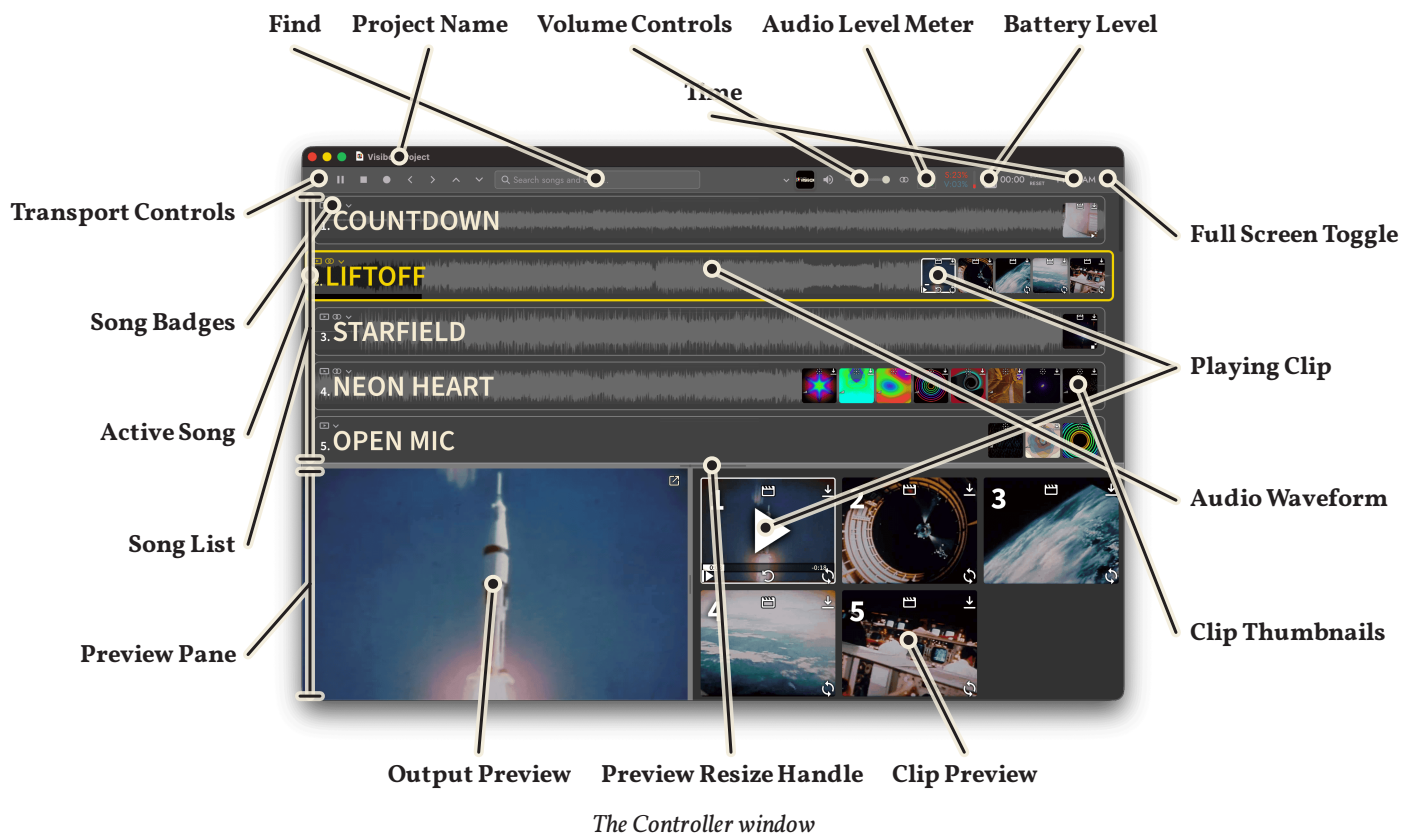
Opening a Visibox Project gives you two windows: the Controller window and the Output window. Here is what is in each.



The Controller window, left, and the Output window, right.

Controller window

The Controller window is where you build a Project and where you drive it during a show. It holds the Songs and Clips, shows what is playing, and gives you everything you need to change what happens next. In performance it usually sits on a display only you can see.



The Control Bar

The Control Bar runs along the top of the Controller window. It gives you transport controls, readouts, and the information worth having in front of you during a show. Turn it on and off in Settings > General.



The Control Bar

The left of the Control Bar holds the transport buttons:

Button	Description
Play	Starts the first Clip in the selected Song.
Pause	Pauses the playing Clip.
Stop	Stops the playing Clip.
Record	Records the Output to a file. See Record .
Previous Clip	Plays the previous Clip in the selected Song.
Next Clip	Plays the next Clip in the selected Song.
Previous Song	Cues the previous Song in the Project.
Next Song	Cues the next Song in the Project.

Record

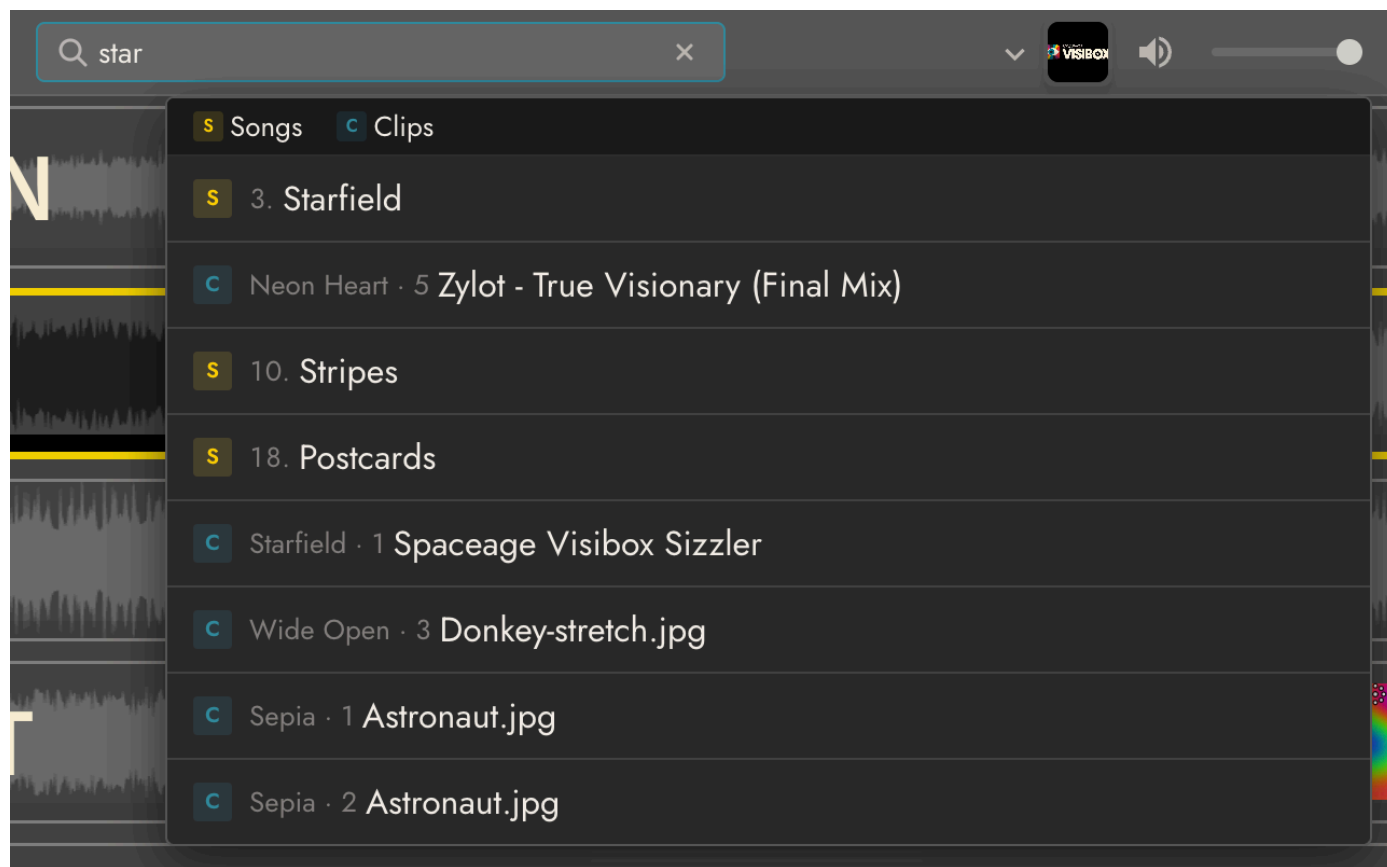
The Record button sits with the transport controls and captures what the Output is showing.

Click it and a small menu offers two ways to start:

- **Start Recording Now** begins immediately.
- **Start Recording on Next Trigger** arms the recording and waits, so the file starts on the first frame of the next Clip you fire rather than on a black stage.

The button lights up while recording is in play: pulsing red once it is armed and waiting, steady red once it is recording. While it records the icon becomes a stop square, and clicking it ends the recording. Clicking an armed button cancels the arming instead of starting anything. Both commands also sit in the Play menu, and **Start Recording Now** carries the R shortcut.

Find



The Find box shows matching Songs and Clips as you type.

The Find box in the center of the Control Bar jumps you to any Song or Clip in the Project. Type a few letters of a Song title or a Clip filename and matches appear as you type. On a large Project it is the fastest way to get somewhere.

To use Find:

1. Click in the Find box, or press F to focus it.

2. Start typing. Results appear as you go.
3. Use the **Up and Down arrow keys** to move between results.
4. Use the **Left and Right arrow keys** to choose an action for the selected result.
5. Press **Enter** to run the selected action, or click an action button directly.

Each result carries action buttons on the right:

For Songs:

Button	Description
Activate	Makes this the active Song. This stops the playing Clip by default; hold Shift to keep it playing.
Trigger	Activates the Song and plays its first Clip.
Show	Scrolls the Song into view and highlights it.

For Clips:

Button	Description
Trigger	Activates the Clip's Song and plays the Clip.
Show	Scrolls the Clip into view and highlights it.

Two buttons sit at the top of the list: **Songs** and **Clips**. Each one turns its own category off. Turn off **Clips** and the list shows Songs only. Turn off **Songs** and it shows Clips only. You cannot turn both off. Visibox remembers the choice.



TIP

Find matches loosely, so you do not need the exact name. Typing “intro” finds “Introduction Video”, “Intro Music”, and anything else with those letters in it.

The right of the Control Bar holds readouts and controls.

Control	Description
Project menu	A small chevron beside the Background Clip. Click it to open the Edit Project panel .
Background Clip	Shows whether the Project has a Background Clip, with a thumbnail and playing status. Click to add, remove, or change it.
Master Volume	Sets the audio level of Visibox.
Imaging	Click to switch the audio between stereo, mono, and multichannel.
Audio Level	A meter for the audio playing in Visibox.

Control	Description
Resource Meter	CPU use for the system and for Visibox. Hover for the detail.
Stopwatch	Click to start, stop, and reset the timer.
Time	The current time of day. Click to adjust the readout.
Battery Level	Battery level and charging status. The Control Bar changes color on battery and flashes when power is low.
Panic Button	Full screen only. Reloads every window if something goes wrong.
Swap Presentation Displays	Full screen only. Click to trade the Controller and Output displays.
Full Screen Button	Enters and leaves full screen.

Tooltips

Hovering an element in the Controller window brings up a tooltip. Clips report their filename, dimensions, duration, frame rate, bitrate, file size, codecs, media type, and Launch Mode. Control Bar elements describe themselves and show live figures, such as current CPU use on the Resource Meter. Tooltips appear immediately, so you can sweep across several elements in a row.

The Song list

The Song list is the body of the window: the Songs in your Project, each with its Clips, in the order you will play them. Drag Songs and Clips to reorder them, and drag a box across the grid to select several at once.

If the Project has more than one running order saved, the Song list shows the one that is active. Choose a different one from [Project > Setlists](#), or open **Manage Setlists...** from the same submenu to build and edit them.

Song badges

Each Song row starts with a small cluster of marks.



Poly Mode, audio imaging, and the Song menu chevron.

- The **Poly Mode** mark shows how this Song shares the stage when several Clips play at once. Every Song has one, including a Song set to Mono, which plays one Clip at a time. Hover it and the tooltip names the mode and what it does.
- The **audio mark** appears on Songs with attached audio and shows whether that audio is mono, stereo, or multichannel.
- The **chevron** opens the Song's editor panel, the same one you get by right-clicking the row.

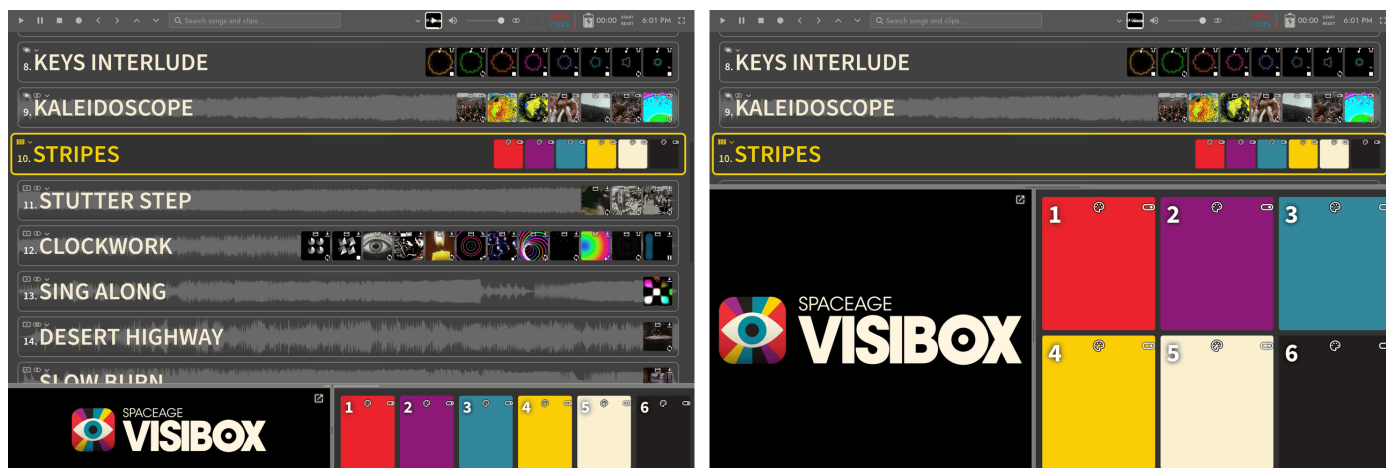
Launch Mode indicators

Each Clip tile carries a small [Launch Mode](#) indicator showing what pressing and releasing that Clip does: start it, hold it, or toggle it. The indicator shows the mode that is actually in effect, which is not always the one stored on the Clip. A Clip left on the default plays as Trigger in a Mono Song but as Toggle in a Song whose Poly Mode lets Clips co-play, and the indicator reflects that rather than the setting.

Hovering the tile spells the mode out in the tooltip.

The Preview Pane

The bottom of the Controller window holds the Preview Pane, split into the Output Preview and the Clip Preview. Drag the handle at the top of the pane to resize it. To turn the pane off, double-click the handle or drag it all the way down.



The same window with a short Preview Pane and with a tall one.

The Output Preview

This shows what the Output is currently displaying. When the Output is on a screen behind you, this is how you see what the audience sees without turning around.

If the Output window is closed or hidden, a button appears in the corner of the Output Preview to open it. It disappears once the window is open.

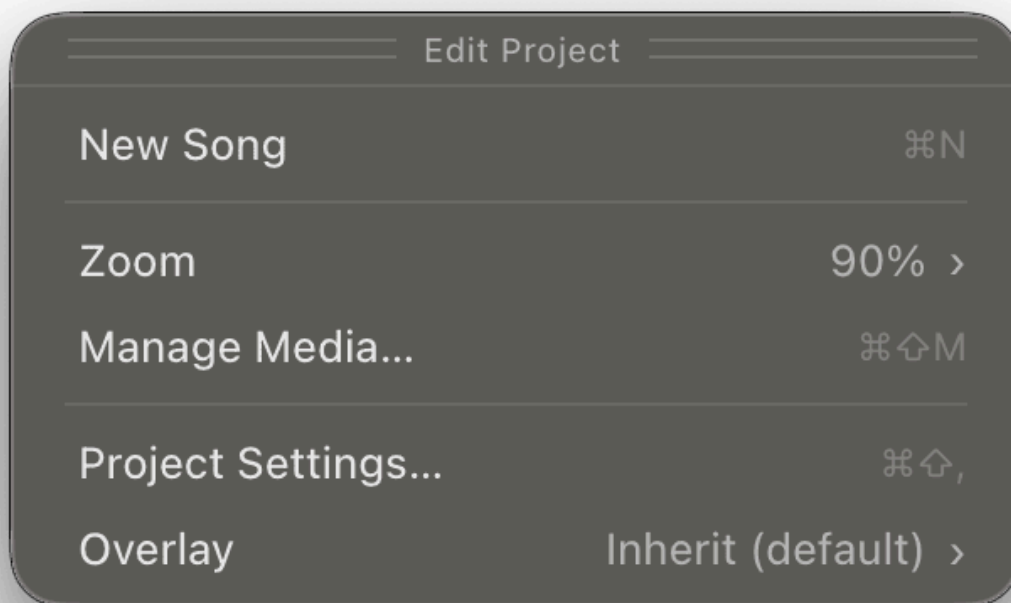
The Clip Preview

This is a closer look at the Clips in the active Song. It is not a timeline: any of these Clips can be fired at any time. The numbers on the thumbnails are the keys that trigger them, and the icons show what happens when a Clip ends or is retriggered (covered in [Clip End Behaviors](#) and [Clip Retrigger Behaviors](#)).

Clicking the progress bar on a playing video Clip scrubs through it.

The Edit Project panel

Right-clicking the empty background of the Song list opens the Edit Project panel, which holds the settings that apply to the Project as a whole. The **Project menu** chevron in the Control Bar, to the left of the Background Clip, opens the same panel with a click.



Right-click the background of the Song list for the Edit Project panel.

Row	Description
New Song	Adds an empty Song to the Project. ⌘N
Zoom	A slider that scales the Song list. The value reads out as a percentage and snaps to 100%.
Manage Media...	Opens and closes the Media window, where you review and optimize the Project's media. ⌘⇧M
Project Settings...	Opens the Project Settings window. ⌘⇧,
Overlay	Assigns an Overlay to the whole Project. Songs and Clips inherit it unless they set their own.

Output window

The Output window shows what Visibox is playing. It is the destination you get without setting anything up, and while you build a Project it is the easiest way to watch your work beside the Controller.

It is one destination of several. Visibox paints a single picture, the output canvas, and hands slices of it to whatever you have switched on: a projector, a second projector, a video wall, an NDI stream, a DeckLink card, Syphon or Spout, and a recording. The Output window takes the same picture they all take. See [Output, Canvas, and Displays](#) for the rest of them.

While you are working you can resize the window to whatever suits you. The size of the window is only the size of the window, and does not change the picture Visibox renders. In performance the Output usually goes full screen on its own display rather than staying in this window, and it takes the shape of that display. What you see now may not be the shape you get then.



TIP

The Output window is black when nothing is playing, so it is easy to miss when you first open a Project.



The Output window displaying a Clip

You can shape the Output to suit the room. Flip or rotate it for rear projection, a ceiling mount, or a vertical screen. Control cropping with Fit Mode and aspect ratio. Spread it across several screens for a video wall. Those settings live in the Settings window.

With the Output Preview showing in the Controller, you may not want the Output window open while you edit. Close it and reopen it later from the Window menu, from the keyboard shortcut, or from the button in the corner of the Output Preview.

Working with Projects

Visibox saves your Project as you work, so there is nothing to remember to do. It also writes [backups](#) as it goes, so a damaged Project file can be restored. Undo and redo let you step back through your changes.

On Macs, the Project data file and its media live inside one self-contained Project file (.vsbx) that you can move between computers or drives. On Windows, the data file (.vsbp) and the media sit together in a folder. To move a Windows Project, copy the whole folder.

Mac and Windows Projects are interchangeable. A Project made on one platform opens on the other.

Saving

There is no Save command, because there is nothing to save. Visibox writes every change to disk as you make it, in a way that cannot leave a half-finished file behind if the power goes out.

That also means there is no recovered-Project prompt after a crash. Other software offers you one because it saves on a timer and the last few minutes are at risk. Visibox reopens the Project exactly as you left it, because the file on disk was already up to date.

If a save ever does fail, Visibox tells you rather than carrying on quietly. The usual cause is a full disk, which gets a message of its own.

Taking a change back

No Save command also means no “revert to last saved”, so undo is how you take something back. Visibox remembers your last forty changes, and each one is a step you can walk back with Z and forward again with Z. The [Edit menu](#) names the step it is about to undo, so the row reads **Undo New Song** or **Undo Move Clip** rather than just **Undo**. Try a setting you are unsure of: you can always walk it back.

Backups

While a Project is open, Visibox writes a complete copy of it every five minutes, as long as something has changed. It keeps the twenty most recent and deletes the rest. They live in a **Backups** folder inside the Project, alongside the media.

What the backups are for

They are insurance against a damaged Project file, and Visibox uses them for you. If the Project file will not open, Visibox notices as it tries and offers you a button: **Open Latest Backup**. Take it. Visibox puts that backup in place of the damaged file and opens it.

That is the whole of it. Backups are a safety net against a damaged file, not a version history: Visibox offers you the newest one and nothing else. To take a change back, use undo instead. Leave the rest of the Project's folder alone, because moving files around in there is the reliable way to break a Project.

NOTE

The backups are copies of the Project's settings and arrangement, not of your media. The media files themselves live in the Project and are not duplicated, which is why twenty backups take up almost no space. It also means a backup cannot bring back media you deleted from the Project.

Setting Up Your Project

Visibox is not video editing software, but it takes enough of the edge off your media that you do not have to prepare files before you use them. It was built to add ambient, atmospheric visuals behind a live performance, though people use it for plenty of other things.

Here is how a show goes together.

Gather your visuals

Start by collecting the video and images you want on screen. These formats play natively:

Video:

- MP4 / MPEG4 (mp4, m4v, mpeg, mpg, mpv, m4p)
- QuickTime (mov, qt)
- Ogg (ogv)
- WebM (webm)

The container is not the whole story: a few codecs need converting even inside these containers. HEVC (H.265) video is the common one. On the systems that cannot play it directly, Visibox keeps your original file and prepares a playable copy in the background. See [HEVC proxies](#).

Video conversion: Visibox tries to convert anything else into a native format on import. If it cannot, it tells you.

Images:

- JPEG (jpg, jpeg, pjpeg, jpj)
- PNG (png, apng)
- WebP (webp)
- GIF (gif)
- BMP (bmp)
- SVG (svg)

Animated GIF files are converted to video, which cuts their size and gives you more control over playback.

Mixed resolutions and aspect ratios are fine. Visibox makes them work.

Creating a new Project

Launch Visibox and choose **File > New Project** (). Every Clip you add is stored with the Project.

DANGER

Do not rearrange files

On Macs, the Project data file and media files are kept in a self-contained Project package file (.vsbx) that can be moved from one computer or drive to another. The package contents are not easily accessible, but they hold all of the media you add.

On Windows, the data file (.vsbp) and media files are stored together in a folder. Do not rearrange the files in it, especially those in the “Files” subfolder. Use Visibox’s **Clean Up Unused Files...** to delete media you no longer need. Do not delete or move files with Finder or File Explorer. That will break your Project.

To move a Windows Project to another computer or drive, copy the entire folder.

Add a Background Clip

With nothing playing, the Output window shows black. A Background Clip replaces that black with whatever you choose, so the audience never sees a blank screen between Songs.

A Background Clip can be any kind of Clip:

- **A video or an image**, in any of the formats listed above. Animated GIFs are converted to video on import, the same as anywhere else.
- **A camera**, so a live feed fills the gaps between Songs.
- **A visualizer**, which reacts to the room and never repeats.
- **A color**, for a flat wash behind everything.

Setting one

Project > Add Background Clip opens a submenu with a row per type:

- **Image or Video...** opens a file dialog.
- **Camera** lists every connected camera and video input.
- **Visualizer** lists your Visualizer presets.
- **Color** lists the preset color swatches.

Once a Background Clip is set, the row reads **Replace Background Clip** and does the same job.

Two shortcuts do the same thing without the menu:

- **Click the Background Clip thumbnail in the Control Bar** at the top of the Controller window. The thumbnail shows at a glance whether a Background Clip is set and whether it is playing, and clicking it lets you add, replace, or remove one mid-show.

- **Drag an image or video from your desktop onto the Output window.**

Adjusting and removing it

Project > Edit Background... opens the Background Clip's own settings: the same playback, fit, transform, and effect controls any Clip has.

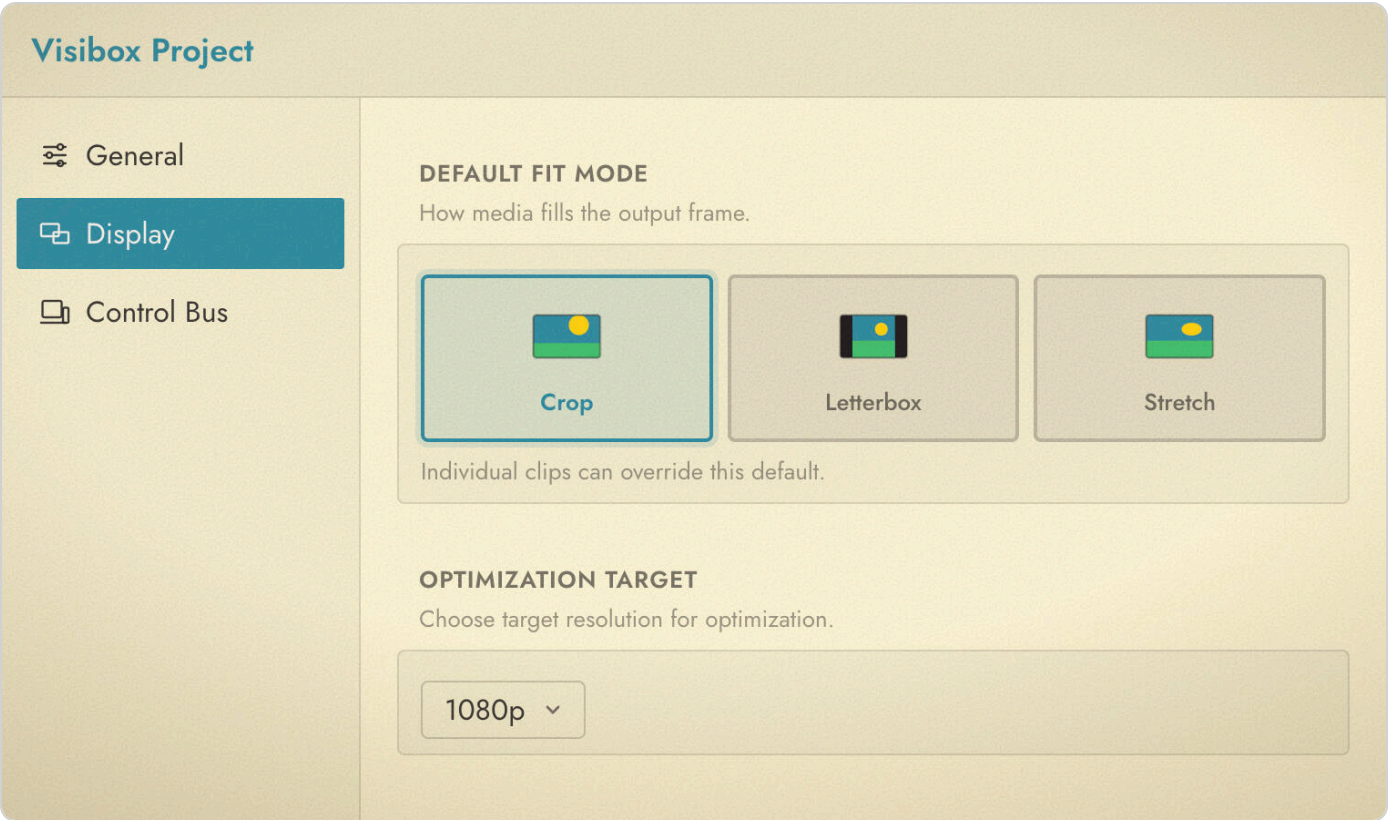
Project > Remove Background Clip... takes it away and puts the black screen back.

The Background Clip belongs to the Project as a whole. You cannot set a different one per Song. If your Clips fade in and out, they cross-fade to and from the Background Clip.

For the menu rows themselves, see the [Project Menu](#) reference.

Project Settings

Project Settings holds the handful of settings that belong to one Project rather than to the app. Open it with **Project > Project Settings...** ().



The Project Settings window. Categories are on the left, and the Project the window is editing is named across the top.

Everything about how the Output itself looks (its size, rotation, flip, enforced shape, rounded corners and multi-display layout) is app-wide and lives in the Settings window instead. See [Output Settings](#) and [Canvas Settings](#).

Choosing which Project you are editing

With one Project open, the window edits that Project and names it across the top.

With several Projects open, a picker appears there instead. **Follow active Project** keeps the window pointed at whichever Project you are working in, so bringing a different Controller window forward re-points Project Settings with it. Turn it off to pin the window to one Project and pick that Project by hand.

General

Setting	What it does
Project Title	The name shown for the Project in Visibox. Leave it blank to use the Project folder's own name.

Setting	What it does
Uppercase Titles	Shows Song and Clip titles in the Controller in uppercase. Turn it off to see titles exactly as you typed them.

Display

Default Fit Mode decides how media fills the Output frame for every Clip in the Project that has not been given its own answer.

Mode	What it does
Crop	Fills the frame edge to edge, cropping whatever does not fit. This is the default.
Letterbox	Shows the whole Clip, adding bars to make up the difference in shape.
Stretch	Fills the frame completely by distorting the Clip to match it.

Individual Clips can override the default from the Clip editor's own Fit Mode row.

Control Bus

The Control Bus category is where a Project's named Controls are defined and bound to the Clip and Effect parameters they drive. Every Project starts with eight, lettered A to H, and you can add more. It is documented on the [Real-Time Control](#) page rather than here, because the setup only makes sense alongside the controls it feeds.

Adding Songs

When you create a Project, it comes with one Song already in it.

The quickest way to add another is . You can also use **Project > New Song**, or right-click the Controller window background and choose New Song.

To rename a Song, double-click its title and start typing. **Song > Edit Song Title...** does the same thing.

NOTE

A “Song” is the container Visibox uses for visual Clips, because Visibox was built with musical performance in mind. If you are running something other than a band, a Song can just as easily be a scene, a service, an act, or an exhibit.

How the Song list looks

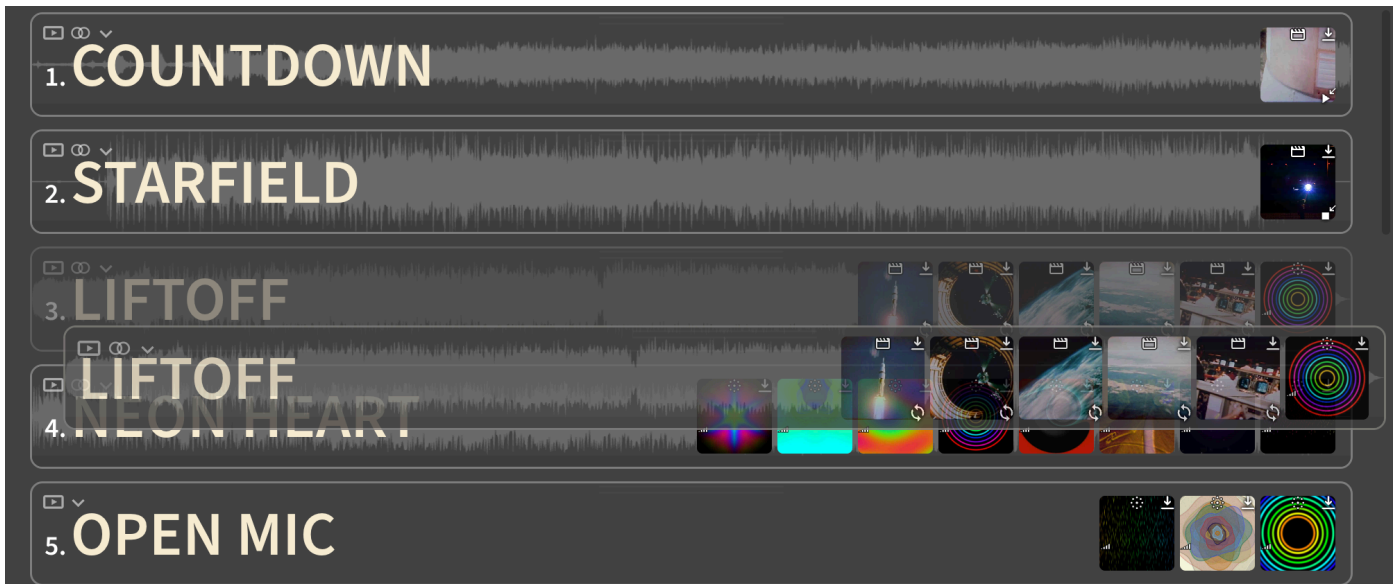
New Songs get random spaceage-flavored names. If you would rather they all arrive as “New Song”, turn on **Simple Titles for New Songs** under [Settings > General](#).

Song titles are uppercase by default. **Uppercase Titles** in the [Project Settings](#) window () turns that off and shows titles the way you typed them.

The Song list text is deliberately big, so you can read it from across a stage. Zoom the whole Controller window with  and  , or from the Project menu.

Rearranging Songs

Grab the handle at the top of a Song frame and drag it to a new position. The Song numbering updates as you go. If you prefer the keyboard, select a Song and use  and .



Grab the handle at the top of the Song frame to drag the Song into a new position.

Reordering changes the running order of the active Setlist. A Project can hold several running orders at once. See [Setlists](#).

Duplicating Songs

⌘D makes a copy of the selected Songs, along with their Clips and settings. **Song > Duplicate Song** does the same.

You can also **hold** **⌘** **and drag** a Song by its handle. The Song is copied to where you drop it instead of moved. The cursor changes to a copy cursor as soon as you hold the key, so you can tell which gesture you are about to make before the drag starts.

Duplicates point at the same media files as the original, so a copy costs almost no disk space.

Cutting, copying, and pasting Songs

Songs go on the clipboard like anything else.

- **⌘X** cuts the selected Songs.
- **⌘C** copies them.
- **⌘V** pastes them into the Project.

The Edit menu names what it is about to do, so the rows read **Cut Song**, **Copy 3 Songs**, **Paste 2 Songs**, and so on. The same three actions are on the Song right-click menu. See the [Edit Menu](#) reference.

Paste works across Projects. Open both, copy in one, paste in the other, and the media files travel with the Songs. Large videos take a moment to copy.

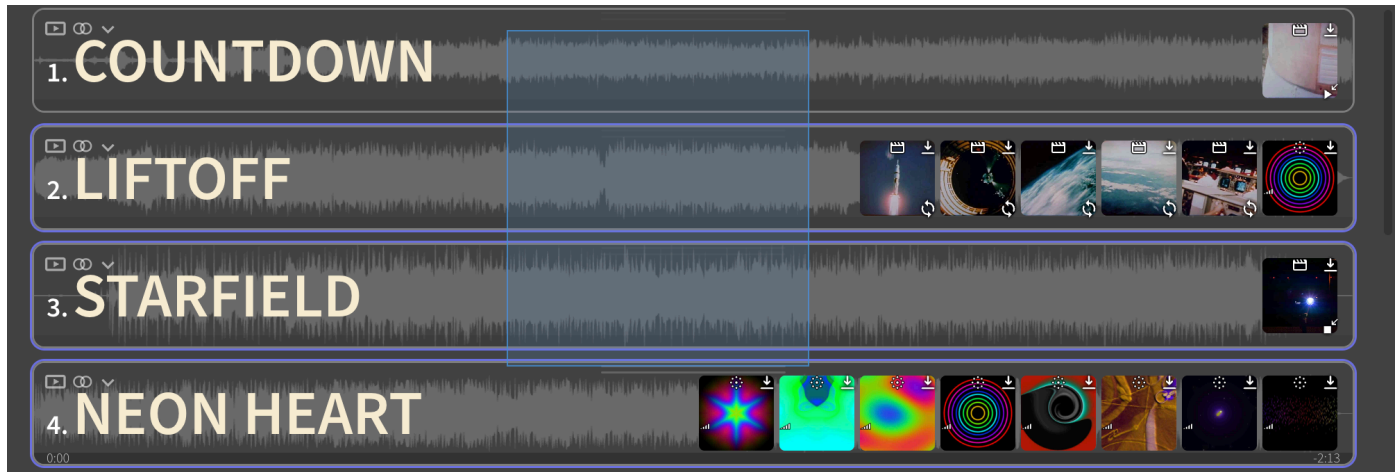
You can also drag a Song by its handle straight from one Project's Controller window into another. That copies the Song with all of its Clips and settings.

Selecting several Songs at once

Click in the empty space beside the Song handles and drag a selection rectangle across the handles you want. Selected Songs are outlined in blue.

- Hold  while you drag to add to the selection.
- Hold  while you drag to remove from it.
- Hold  (Mac) or  (Windows) and click a handle to toggle that one Song.

The list scrolls on its own when you drag past the top or bottom edge, so a long Project does not fight you.



Draw a selection box around the Song handles to select several Songs.

With several Songs selected you can drag, duplicate, cut, copy, and delete them together. You can also edit the options they have in common. If every selected Song has audio attached, changing the volume changes all of them. If some do not, the volume control is unavailable.

Attaching audio to a Song

You can attach an audio file to a Song as a backing track that plays under whatever Clip is running. These formats play natively:

- MP3 (mp3)
- AAC (aac, m4a)
- WAV (wav)
- AIFF (aif, aiff)
- FLAC (flac)
- Ogg (ogg, oga)
- Opus (opus)

Anything else, Visibox tries to convert to a native format on import. If it cannot, it tells you.

NOTE

You are most likely playing the song itself, so attached audio is for something extra. You do not need to attach a recording of the music you are about to perform.

To attach a track, right-click the Song and choose **Attach Audio...**, or use . Once a Song has audio, that row reads **Replace Audio...** instead, and **Remove Audio...** becomes available.

A Song can hold one audio track. If your videos have their own audio, the Clip audio and the backing track both play.

Attached track, or Audio Clip

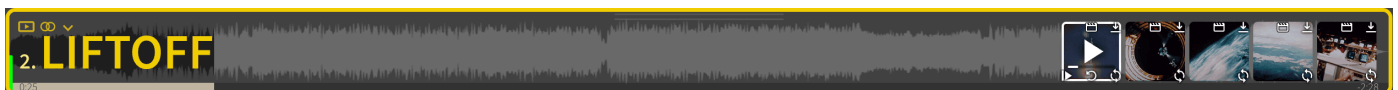
An audio file can be either. A Song has one attached track, playing under everything; it can hold any number of [Audio Clips](#), each triggered on its own. Attach the track you are playing along to, and use Clips for the stingers, effects and beds you fire by hand.

Because both are right answers, dropping a single audio file on a Song asks which you meant:

- **Attach to Song** makes it the backing track.
- **Create Audio Clip** makes it a Clip.
- **Cancel** does nothing.

Song > Add Audio Clip... never asks. It always makes Clips, and takes several files at once.

Dropping several audio files at once never asks either. Visibox offers to combine them as one multichannel backing track instead.



A Song with background audio playing.

While the audio plays, the Song frame shows its waveform, a progress bar with elapsed and remaining timers, and a green level meter down the left edge. A meter on the Song means the sound is coming from the backing track rather than from a Clip. See [Audio Metering](#).

Drag the seek bar below the waveform to scrub to any point in the track.

Mute, volume, and what happens when the track ends are all in the Song Editor Panel. Right-click the Song and choose **Edit Song....** See [Song Editor Panel](#).

Setlists

A Setlist is a saved running order of the Songs in a Project. One Project can hold as many Setlists as you like, so a single Project can carry your headline set, your acoustic set, and the wedding version of the same material without duplicating anything.

The Songs live in the Project. A Setlist only says which of them appear, and in what order.

The active Setlist

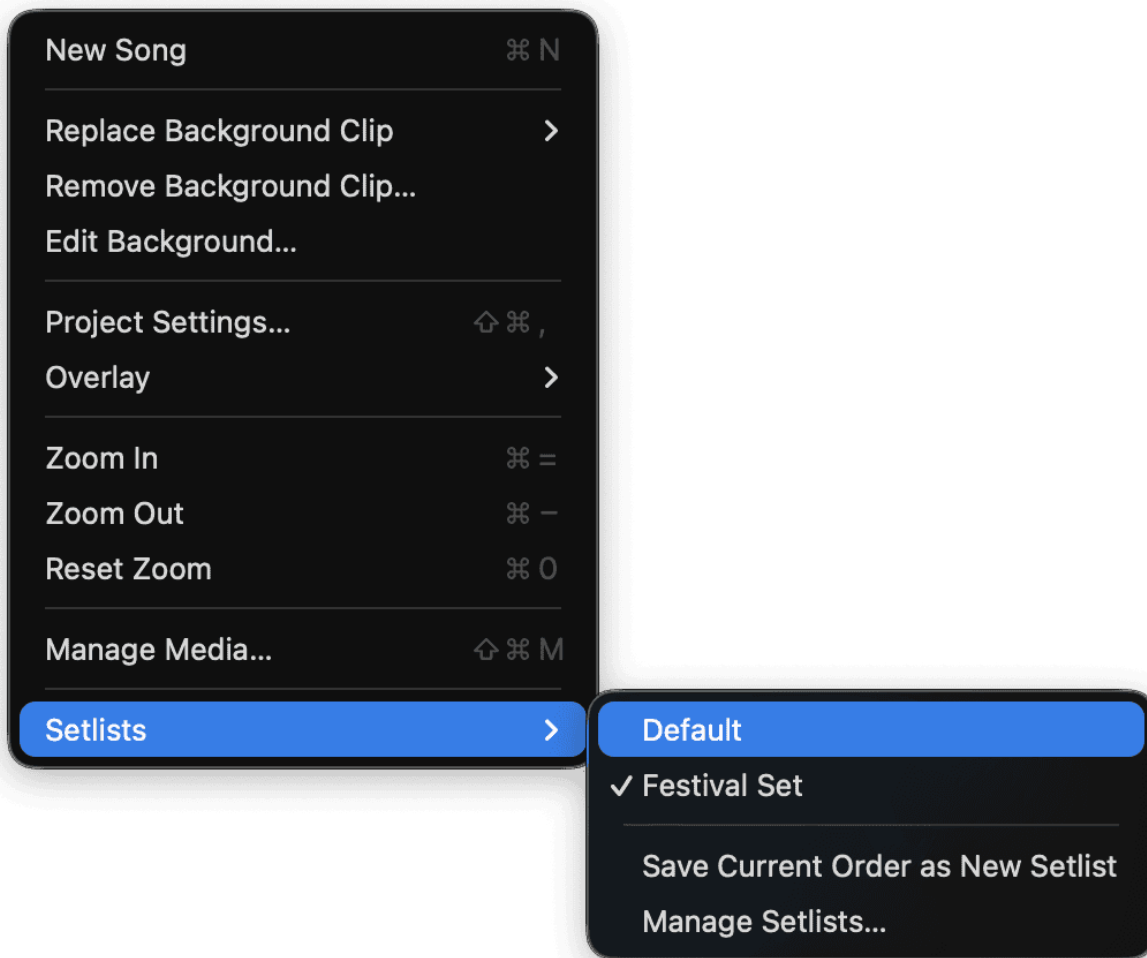
One Setlist is always active. The Controller window's Song list *is* the active Setlist: the Songs in it, in its order. Switch Setlists and the Song list changes to match.

Songs left out of the active Setlist are still in the Project. They are hidden from the Controller until you add them back or switch to a Setlist that includes them.

Every Project has at least one Setlist. A Project made in an earlier version of Visibox gets one named **Default** the first time you open it, holding the Song order it already had.

Switching Setlists

Open **Project > Setlists**. Each Setlist in the Project appears as a radio item at the top of the submenu. Choose one to make it active.



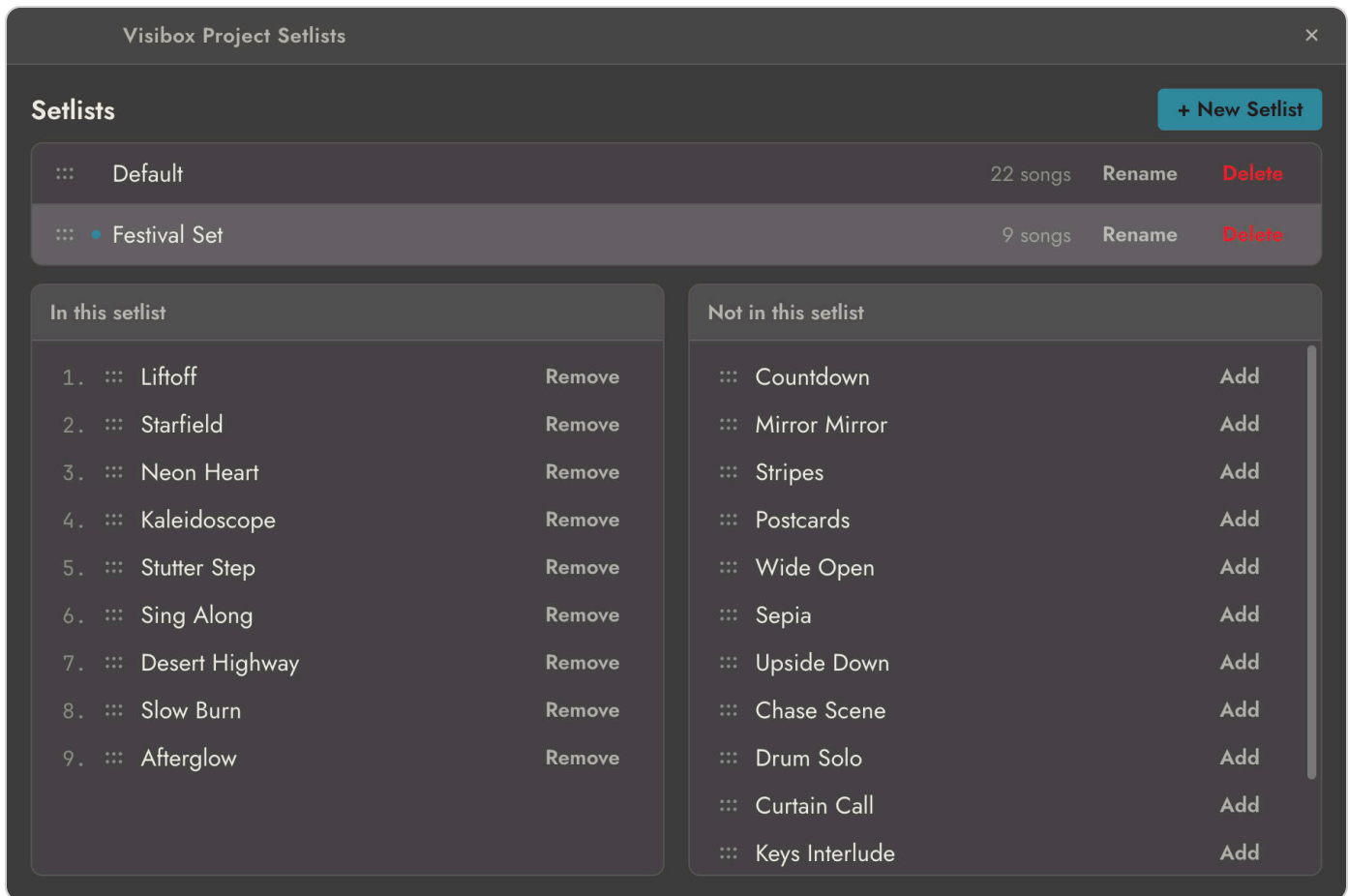
Project > Setlists lists every Setlist in the Project. The checked one is active.

Two actions sit below the list:

- **Save Current Order as New Setlist** takes the Song list exactly as it stands and saves it as a new Setlist, then makes it active. Visibox names it for you (Setlist 2, Setlist 3, and so on). It is the fastest way to keep a running order you have finished dragging into shape.
- **Manage Setlists...** opens the Setlists window.

The Setlists window

Project > Setlists > Manage Setlists... opens a window titled with your Project name followed by “Setlists”. Everything about Setlists that is not a straight switch happens here.



The Setlists window. The Setlist list on the left, the active Setlist's Songs in the middle, and the rest of the Project's Songs on the right.

The window opens centered over the Controller window. Press  or click the close button to dismiss it.

Managing the Setlists themselves

The list of Setlists shows each one with the number of Songs it holds. The active Setlist is marked.

- **Click a Setlist** to activate it. The Song columns update immediately.
- **+ New Setlist** creates a Setlist from the current Song order and drops straight into renaming it.
- **Rename** puts the name into an edit field. Press  to keep the new name or  to cancel. You can also double-click the name.
- **Delete** removes the arrangement. Visibox asks first, and the confirmation reminds you that the Songs themselves stay in the Project. You can take it back with **Edit > Undo**.
- **Drag a Setlist** by its grip to change the order of the list.

Visibox will not delete your last Setlist. When only one is left, the Delete button is unavailable, and hovering it explains why.

Adding and removing Songs

The two columns to the right of the Setlist list are **In this setlist** and **Not in this setlist**. Between them they account for every Song in the Project.

- **Drag a Song from right to left** to add it to the active Setlist. Drop it where you want it to land in the running order.
- **Drag a Song from left to right** to take it out of the Setlist. The Song stays in the Project.
- **Drag within the left column** to reorder the running order.
- **Add** and **Remove** buttons on each row do the same thing without dragging. Add appends to the end of the Setlist.

Drag across several Songs to select them as a block, then drag the block between the columns in one go. The columns scroll as you drag past their edges, so a long Song list does not need two hands. A block move counts as one step, so **Edit > Undo** puts the whole selection back at once.

When every Song in the Project is already in the active Setlist, the right column reads “Every song in the project is in this setlist.”

Setlists and the rest of the Project

- **Adding a Song** to the Project adds it to the active Setlist. Songs you add while one Setlist is active are not automatically in the others.
- **Reordering Songs** in the Controller window (dragging a Song frame, or  and ) changes the active Setlist’s order. See [Adding Songs](#).
- **Deleting a Song** removes it from the Project and from every Setlist that used it. To take a Song out of one running order without losing it, remove it from the Setlist instead.



TIP

Build one Project holding everything you play, then make a Setlist per night. Reordering on the fly during a show changes only the Setlist you are in, so last night’s running order is still there in the morning.

Adding Clips

Once a Project has Songs in it, you can start filling those Songs with Clips. A Clip is one piece of content you can trigger: a video, an image, a live camera, a Visualizer, a solid color, or an audio file.

This page covers getting Clips into a Project and arranging them. What a Clip does once it plays is on [Clip Playback Options](#), [Launch Modes](#) and [Poly Mode](#).

Adding a Clip

The quickest way is to drag a video or image file from your desktop straight into a Song row. The Clip's thumbnail appears in the row as soon as it lands.

There are four other ways in:

- Select files in the Finder or File Explorer, copy them, then select the target Song in Visibox and paste.
- Choose **Song > Add Media Files...**, or right-click the Song and choose the same item.
- Choose **Song > Add Audio Clip...** to add sound files as Clips.
- Press  with a Song selected.

Drop a single audio file on a Song and Visibox asks what you meant by it, because there are now two right answers. See [Audio Clips](#).

Clip types

Not every Clip comes from a file.

-   **Video and image Clips** come from media you add. These are the ones you drag in.
-  **Camera Clips** show a live video input: a USB webcam, an HDMI capture device, or a virtual camera. See [Connecting Cameras & Video Inputs](#).
-  **Visualizer Clips** render live audio-reactive graphics rather than playing a file. See [Visualizers](#).
-  **Color Clips** are a solid color. See below.
-  **Audio Clips** play a sound file. See below.

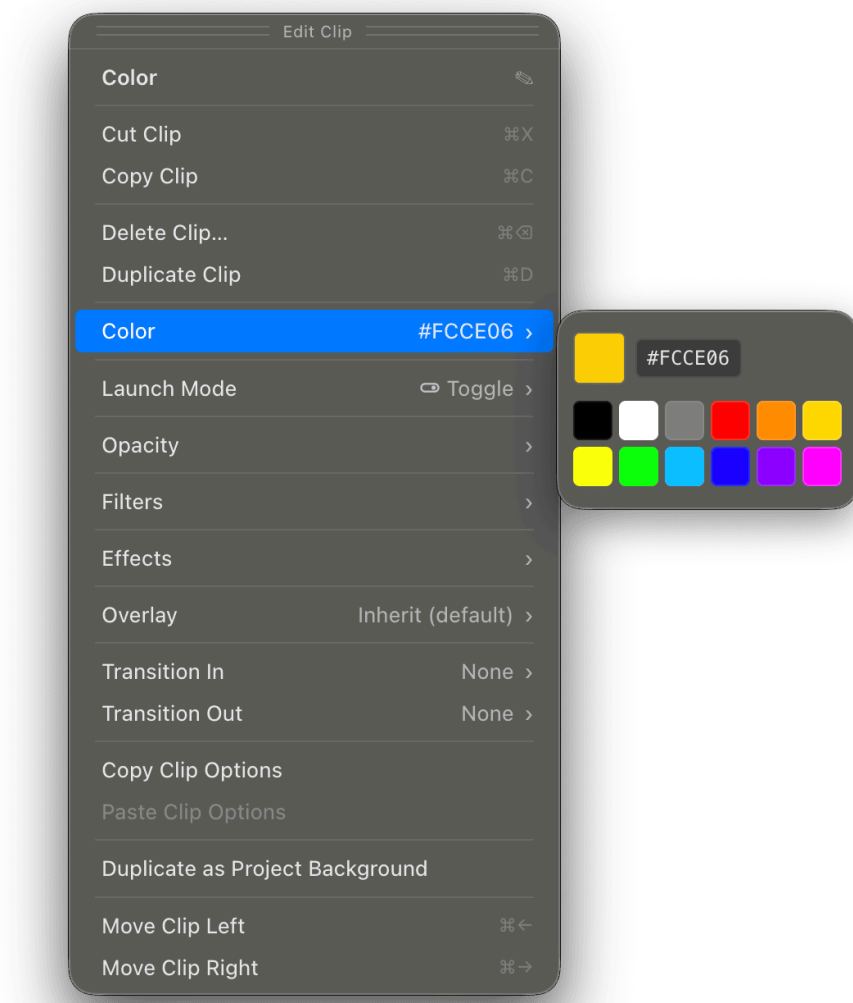
Each Clip tile carries its type's icon at the top of the thumbnail, and the hover tooltip leads with the same icon, so you can tell a Camera Clip from a video Clip across the row.

Right-click a Song and choose **Add Camera Clip**, **Add Visualizer Clip**, **Add Color Clip** or **Add Audio Clip...** to create one of the last four.

Color Clips

A Color Clip is a solid color, and it replaces the placeholder Clips from older versions. Visibox has to be playing something to play a Song's audio, so a black Color Clip is how you play nothing visually. That is what you want for a Song that is only a backing track. An [Audio Clip](#) is the other way to get sound with nothing on screen, and it is black on its own without needing a Color Clip under it.

To add one, right-click a Song and choose **Add Color Clip**, or use the Song menu. A color picker opens.



Pick any color for a Color Clip.

Color Clips earn their place well beyond blackouts:

- **Blackouts.** A black Color Clip is a clean cut to nothing between songs or sections.
- **Backgrounds.** A deep blue or a warm amber sets a tone without any footage.
- **Canvases.** Apply an Effect to a Color Clip and you get texture and motion out of nothing. A white Clip with a glitch or film-burn Effect holds up on stage.

TIP

A dark Color Clip with an audio-reactive Effect on it is a fast way to get a minimal, responsive look with no media files at all. See [Effects](#).

Audio Clips

An Audio Clip plays a sound file, and it is a Clip like any other: it sits in a Song, it has a number key, it has a Launch Mode, and it takes Effects. A Song can hold as many as you like.

That makes it the piece a Song's single attached backing track cannot be. Use one for a stinger you fire on cue, an intro bed you leave running, a sound effect, or a second track you drop in over the first.

To add one, choose **Song > Add Audio Clip...**, or right-click the Song and choose the same item. Pick as many files as you like and each becomes its own Clip, named after its file.

Attach to the Song, or make a Clip

Drop a single audio file on a Song and Visibox asks whether to attach it as the Song's backing track or create an Audio Clip from it. See [Attached track, or Audio Clip](#) for the choice and what dropping several files at once does instead.

What the audience sees

An Audio Clip has no picture of its own, so it plays black by default. You have two ways to give it one, and they are alternatives rather than layers:

- **A color.** Pick one in the Clip Editor Panel and the Clip plays as that solid color.
- **A Visualizer.** Pick one and it draws live from the audio, exactly as a Visualizer Clip does. Choosing a Visualizer clears the color, and choosing a color clears the Visualizer.

Paired with a Visualizer, an Audio Clip gets the Fit Mode, Flip and Rotate rows that a picture needs. On its own it behaves like a Color Clip and does not.

In the Controller, an Audio Clip's thumbnail shows the file's waveform until you pair a Visualizer with it, at which point it shows the Visualizer instead.

What an Audio Clip cannot do

Everything that plays a Clip backwards needs a reversed copy of the media, and Visibox only makes those for video. So on an Audio Clip:

- **Direction** is not offered at all. Select an Audio Clip together with video Clips and the row disappears for the whole selection.
- **End Action** has no Bounce.
- **Retrigger Action** has no Reverse and no Scratch.
- **Resume Action** has no Sync to Song Position, which needs video frames to follow.
- **Loop Crossfade** is video only, for the same reason as Bounce.

Everything else is there: trim points, Speed, Launch Mode, Mute, Volume, Velocity Sensitivity, Filters, Effects, Overlay and Transitions.

Reacting to its own sound

Pair an Audio Clip with a Visualizer and an **Audio Source** row appears, with a value the other Clip types do not have: **This Clip**. It makes the Visualizer react to that Clip's own sound rather than to everything Visibox is playing, so two Audio Clips running at once each drive their own picture.

It reads the Clip's level after its Volume, so turning the Clip down calms the Visualizer with it. New Audio Clips start on This Clip. See [Audio Source](#).

Arranging Clips

Drag a Clip thumbnail to a new position to reorder it.  and  move the selected Clip left and right. You can also drag a Clip from one Song into another.

Clip tiles play on press, so dragging a tile from anywhere on its face would start it. Hover a tile in the Song list and a grip appears along its top edge; drag from the grip to move the Clip. Clips in the Preview Pane cannot be dragged while Trigger Clips on Press is on. See [Launch Modes](#) for the setting that controls this.

To duplicate a Clip, right-click it and choose **Duplicate Clip**, or press . Duplicates can be dragged into other Songs. All duplicates of a Clip share one media file on disk, so they cost no extra space.

Cut, copy and paste

Clips can be cut and copied like any other content. Select one or more Clips and press  to cut or  to copy. A cut Clip stays where it is, outlined and dimmed, until you paste it somewhere.

Select the Song you want them in and press  to paste. The [Edit menu](#) names the item for what is on the clipboard, so it reads **Paste Clip** or **Paste 3 Clips**.

The same items are in the Song Editor Panel.

Selecting Multiple Clips

Multiple Clips can be selected the way files are selected in the Finder or File Explorer. Click outside a Clip and drag a rectangle around the ones you want. Selected Clips are outlined in blue.

- Hold  and drag again to add more Clips to the selection.
- Hold  and drag to remove Clips from it.
- Hold  (Mac) or  (Windows) and click a Clip to toggle it.



Hold the shift key to add more Clips to an existing selection.

Edit > Select > All Clips ( **A**) takes everything. **Edit > Select > Deselect All** ( **A**) clears the selection, and so does clicking outside any selected Clip in the Controller.

Setting options for several Clips at once

With multiple Clips selected you can drag, duplicate and delete them together, and you can edit any option they have in common. Select five video Clips and one change to **End Action** applies to all five.

Mix Clip types and the shared list gets shorter. Select a video Clip and an image Clip together and End Action is not offered, because an image has no end to reach.



With several Clips selected, right-click to set options for all of them.

Copy Clip Options and **Paste Clip Options** do the same job in two steps. Copy the settings off one Clip you have dialed in, select as many Clips as you like, and paste. Only the settings travel; the media does not.

Copying Clips between Projects

Drag Clips from one Project window to another to copy them. Open both Projects, select the Clips you want, and drag them across. One Clip or a whole selection both work.

The Clips arrive with all of their settings, and their media files come with them, so a large transfer takes a moment.

The Background Clip

A Project can have a Background Clip: a Clip that plays underneath, showing whenever no other Clip is up. It can be a video, an image, a camera, a Visualizer or a color.

To promote a Clip you already have, right-click it and choose **Duplicate as Project Background**. See [Working with Projects](#) for the rest.

Clip audio

A video Clip with an audio track plays that audio when it plays, and an [Audio Clip](#) is nothing but audio. Right-click either and use **Mute** to silence it, or **Volume** to set its level against the rest of your material.

Clip volume and Song volume both go past 100%. The slider stops at 200%, and you can click the value and type a higher number when a Clip is much quieter than everything around it. This is per Clip and per Song, not the output volume.

When a Clip is making sound, a green level meter appears on the left edge of its thumbnail in the Controller. It tells you at a glance which Clip the audio is coming from, which matters when a Song has both a backing track and Clips with audio. See [Audio Metering](#).

A Song's backing track and every Clip playing sound are added together, each at its own volume. Nothing ducks anything else, so an Audio Clip fired over a running backing track plays on top of it.

Clips in the Preview Pane

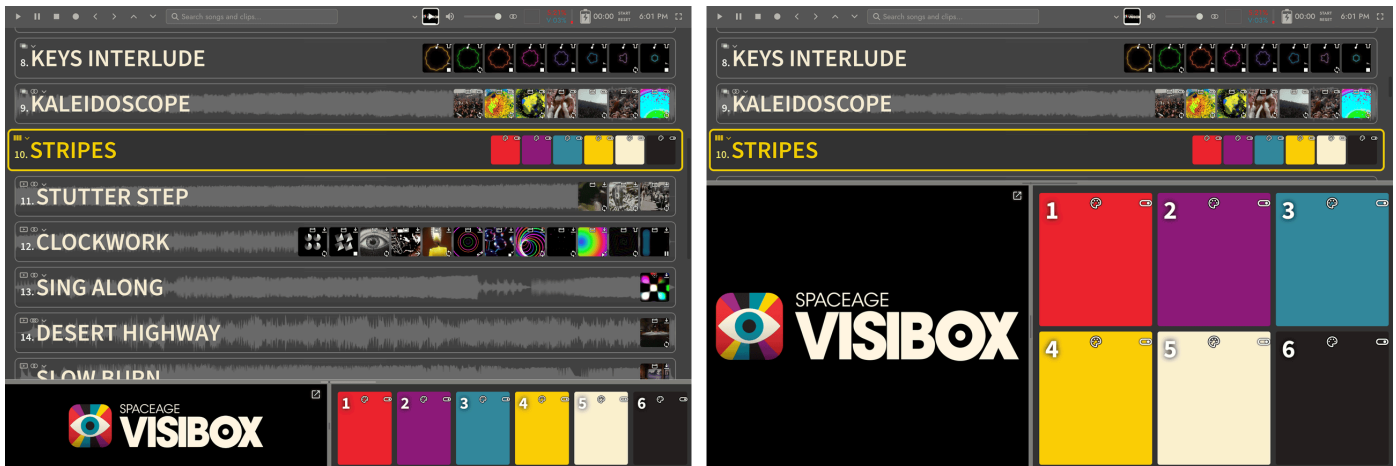
When a Song is active, its Clips appear in the Preview Pane at the bottom of the Controller window. Each Clip carries a number in its top-left corner, which is the key that triggers it. Clips 1 through 9 are on the number keys, and a MIDI controller can be mapped to the same positions.

The Preview is a reference, not a timeline. Any Clip in it can be triggered at any time, in any order.

Drag the bar at the top of the Preview to make it taller or shorter, which helps when you are reading it from across a stage. Click the bar to hide the Preview entirely.

Seeking through Clips

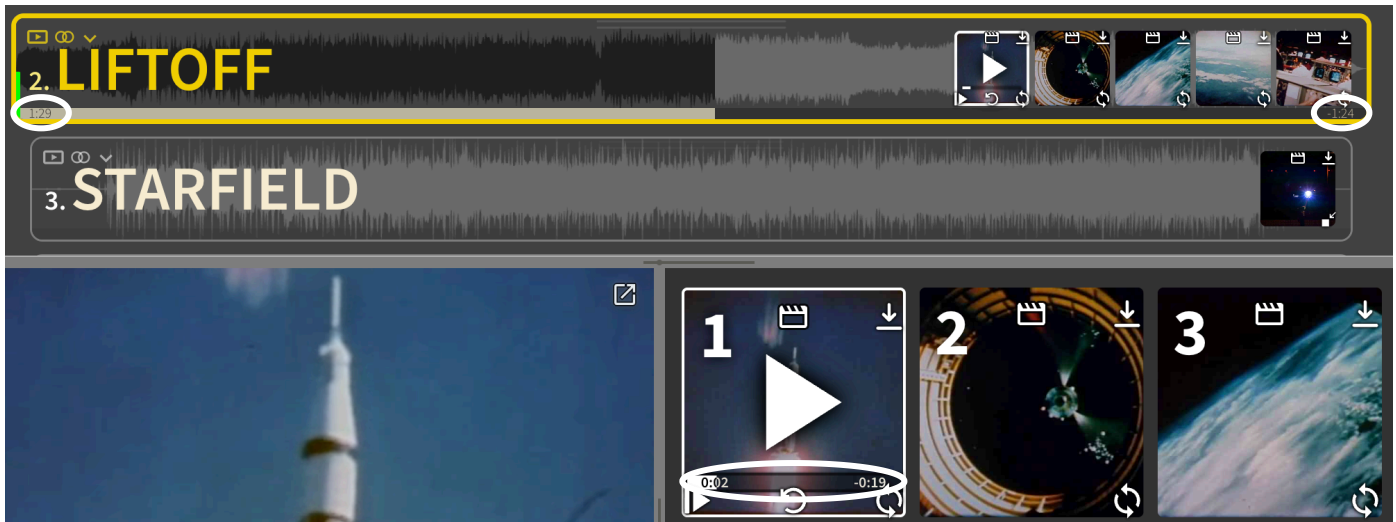
Click a Clip's progress bar in the Preview to jump to a point in it. This is how you find a spot in a long video without playing from the top.



Drag the bar at the top of the Preview to resize it.

Playback Timers

The Controller shows elapsed and remaining time on both Clip and Song progress bars, updating as they play.



Playback timers on the Clip and Song progress bars.

A glance tells you how long is left on a backing track or a video before you need to move.

Playback options

Everything a Clip does once it starts is set in its [Clip Editor Panel](#), and covered on [Clip Playback Options](#). Three of those settings decide most of a Clip's character.

Clip end behaviors

End Action decides what happens when a Clip runs out. The default is Loop, so a Clip keeps playing until you move on. It can also stop, freeze on its last frame, start the next Clip, or move the Song along. An icon in the bottom-right corner of the thumbnail shows the current choice. See [End Action](#).

Clip retriggering behaviors

Retrigger Action decides what a second press does to a Clip that is already playing. Restart is the default; Stutter jumps back a moment, and Do Nothing protects a Clip from an accidental press. See [Retrigger Action](#).

Sync to Song Position

Sync to Song Position locks a Video Clip's position to the Song's audio track, so picture and track stay together however you cut around. It is one of the **Resume Action** choices. See [Sync to Song Position](#).

Try it

Now that you can add and arrange Songs and Clips, take Visibox for a spin.

Download a fully built demo Project from inside the app: **Help > Download/Open Demo Project....** Trigger the first Clip in the first Song and watch it play. The Output window is the easiest place to watch while you learn; if you cannot see it, choose **Window > Output Window** (1).

Then start changing things. Reorder the Songs. Rename a few. Move Clips around and watch what their end behaviors do. Switch a Clip's Fit Mode between Crop and Letterbox to see how Visibox fills the frame. Add a Clip of your own; any video or image you have will do.

Clip Playback Options

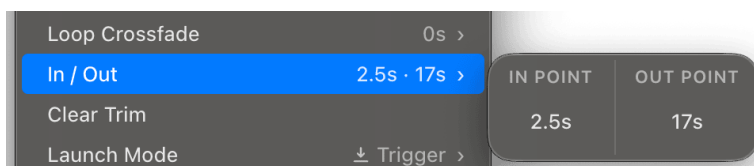
Every Clip carries its own set of playback settings: where it starts, what it does when it reaches the end, how fast it runs, how it arrives on stage, and how it fills the frame. Set them once and they travel with the Clip.

Right-click a Clip thumbnail in the Controller to open its [Clip Editor Panel](#), where all of these live. Select several Clips first and the panel edits every option they have in common at once.

Two shortcuts save a lot of clicking once you have one Clip dialed in. **Copy Clip Options** captures everything on the Clip except its media, and **Paste Clip Options** applies that set to the whole current selection. Both are in the Clip Editor Panel and in the [Clip menu](#).

In and out points

A video Clip does not have to play the whole file. The **In / Out** row picks the section you want, and everything else about the Clip works from there: looping returns to the In Point, and the progress bar measures the trimmed section rather than the file.



In Point and Out Point pick the section of the file the Clip plays.

The row shows both times side by side. Open it and the panel holds an **In Point** field and an **Out Point** field: click a value to type an exact time. Leave a point alone and the Clip uses the natural edge of the file.

There is a faster way to set them while the Clip is playing. Let it run to the frame you want, then choose **Set In Point at Playhead** or **Set Out Point at Playhead**. Both rows appear only while the Clip is playing or paused, and only when a single Clip is selected. **Clear Trim** removes both points and returns the Clip to the full file.

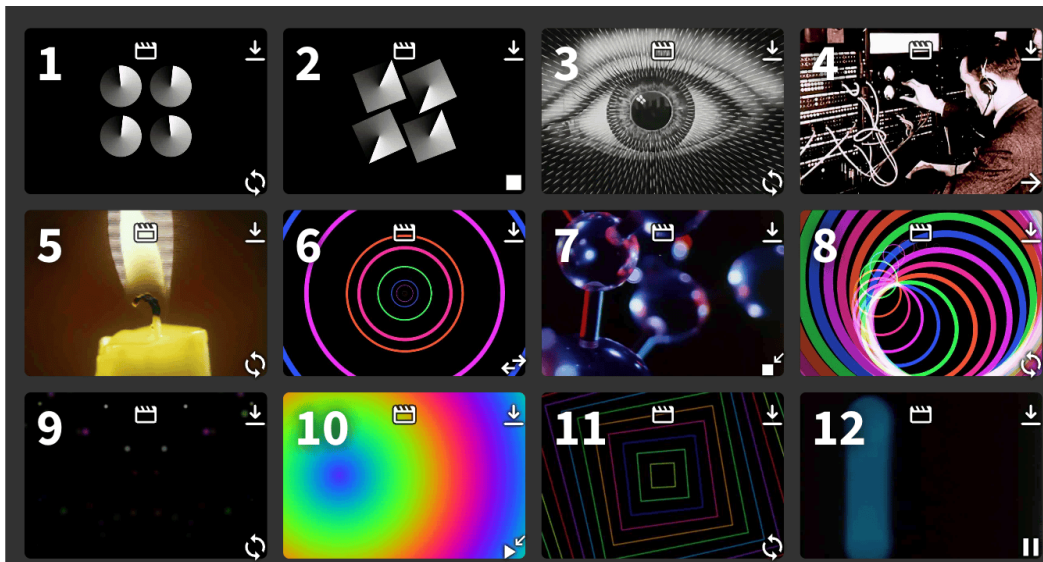
TIP

Trimming is the quickest fix for a downloaded loop with a few dead frames at the head or a logo sting at the tail. Set the points, and the Clip behaves as if the file had been cut.

End Action

End Action decides what happens when the Clip reaches its out point. The default is Loop, so a Clip keeps playing until you move on.

An icon in the bottom-right corner of the thumbnail shows the current choice, in both the Song row and the Preview.



End behavior icons on Clip thumbnails.

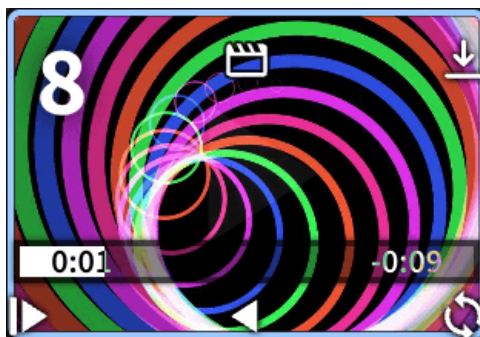
Icon	Name	What it does
	Loop (default)	The Clip starts over.
	Bounce	The Clip reverses direction at each end.
	Stop (One-Shot)	The Clip plays once and stops. The output returns to black or the Background Clip, and the audio stops.
	Freeze	The Clip holds on its last frame.
	Start Next Clip	The next Clip starts playing.
	Cue Next Song	The next Song is activated but does not start.
	Start Next Song	The next Song is activated and starts playing.

NOTE

Stop (One-Shot) is the End Action value: the Clip plays through once and stops on its own. **Stop** is a separate value in Retrigger Action, where it means a second press stops the Clip. The two are different settings with similar names.

Retrigger Action

Retriggering means triggering a Clip that is already playing. **Retrigger Action** decides what that second press does.



The retrigger behavior appears below the progress bar on the playing Clip.

Icon	Name	What it does
▶	Restart (default)	The Clip plays from the beginning.
↺	Stutter	The Clip jumps back about a second.
↻	Scratch	The Clip shuttles back and forth at speed, like a record scratch.
◀	Reverse	The Clip reverses its current direction.
⏸ / ▶	Pause/Resume	The Clip pauses. The next press resumes from that point.
⏸ / ▶	Pause/Restart	The Clip pauses. The next press restarts it.
■	Stop	The Clip stops and the output returns to black or the Background Clip.
No icon.	Do Nothing	The press is ignored. Use this to protect a Clip from an accidental retrigger.

Setting a Clip's [Launch Mode](#) to Toggle overrides this: a second press always stops a Toggle Clip. The Clip Editor Panel grays the Retrigger Action row in that case and shows Stop as the value in effect.

Resume Action

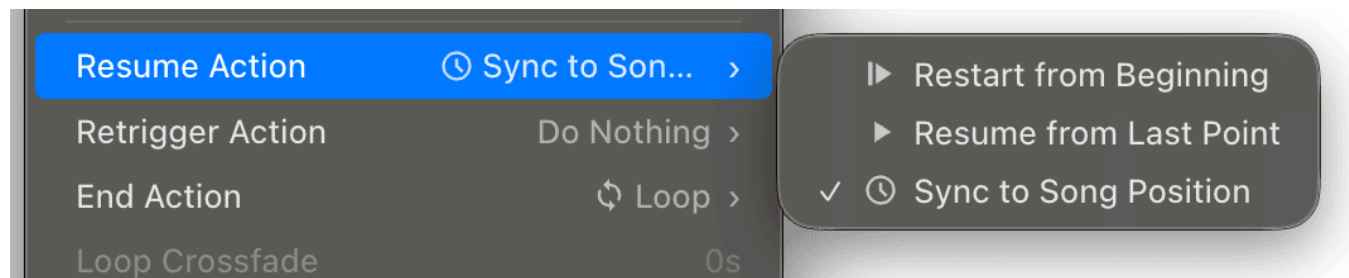
The first time you play a Clip it starts at the beginning. **Resume Action** decides where it picks up after you have played something else and come back.

The choice shows as an icon in the lower-left corner of the Clip thumbnail, visible on hover or while the Clip is active.

Icon	Name	What it does
▶	Restart from Beginning (default)	The Clip plays from the start every time.
▶	Resume from Last Point	The Clip picks up where it left off. Good for a long video you keep cutting away from.
🕒	Sync to Song Position	The Clip stays locked to the Song's audio timeline. See Sync to Song Position .

Sync to Song Position

Sync to Song Position keeps a video Clip's position matched to the Song's audio track at all times. Seek the Song audio to 1:30 and the video jumps to 1:30 with it.



Sync to Song Position is one of the Resume Action choices.

While a Clip is synced:

- The video tracks the Song audio the whole way through.
- You can cut away to an unsynced Clip, such as a camera, and when you cut back the synced Clip is at the position it would have reached had it never stopped.
- Some options are restricted, because the Song is deciding the position rather than the Clip. Retrigger Action, End Action and Direction each fall back to the value that sync allows, and Loop Crossfade is grayed.
- A clock icon (🕒) stays on the thumbnail so you can spot synced Clips at a glance, whether or not you are hovering.

This is what you want for music videos, timed sequences, and anything where the picture has to stay with the track.

Loop Crossfade

Most loops do not wrap cleanly. The last frame and the first frame rarely match, so a looping Clip shows a visible jump every time it comes around.

Loop Crossfade blends the end of the loop into its beginning. Set a duration in seconds and Visibox prepares a version of the Clip whose tail dissolves into its head, so the wrap point stops reading as a cut. A second or two is usually enough.

The row appears only when **End Action** is set to Loop, since a Clip that stops or freezes has no loop point to hide. It is shown but grayed for a Clip playing backward or set to Sync to Song Position, because neither of those can use the prepared version.

Visibox prepares the crossfaded version in the background. Until it is ready, the Clip loops with a hard cut as it always did, then picks up the crossfade on its own.

Speed

Speed changes how fast a video Clip plays without touching the file. The slider runs from 0.0625x to 4x, with snap points at 0.25x, 0.5x, 1x, 2x and 4x. Click the value to type an exact number, including one past the ends of the slider.

Double-click the slider to return the Clip to 1x.

Direction

Direction sets a video Clip to play **Forward** or **Backward**. A Clip playing backward is muted, and it cannot use a Loop Crossfade.

The row is not offered for an [Audio Clip](#), and selecting one alongside video Clips takes the row away for the whole selection.

Fit Mode

Most video for a live show is there to give the stage texture, not to be watched as a film. So by default Visibox crops Clips to fill the Output edge to edge, and you can drop in footage of any shape without editing it first.



The 4:3 source on the left, and the same frame cropped to fill a 16:9 Output on the right.

Fit Mode overrides that per Clip:

- **Default** uses the Project's own Fit Mode, set in the [Project Settings](#) window ().
- **Crop** fills the Output edge to edge and cuts off whatever does not fit.
- **Letterbox** shows the whole Clip, adding black bars to keep its shape.
- **Stretch** fills the Output with no bars, which distorts the shape.

Fit Mode belongs to Clips that carry an image: video, image and camera Clips. Visualizer and Color Clips render to the frame rather than into it, so they have no Fit Mode.



A Clip set to Letterbox.



TIP

Projectors, televisions and monitors rarely match the shape of your footage, so leave Letterbox off unless a Clip has something near its edges you cannot afford to lose. Most background video survives a little cropping, and Crop means you never spend a soundcheck working out why the house projector is 16:10 instead of 16:9.

Rotate and Flip

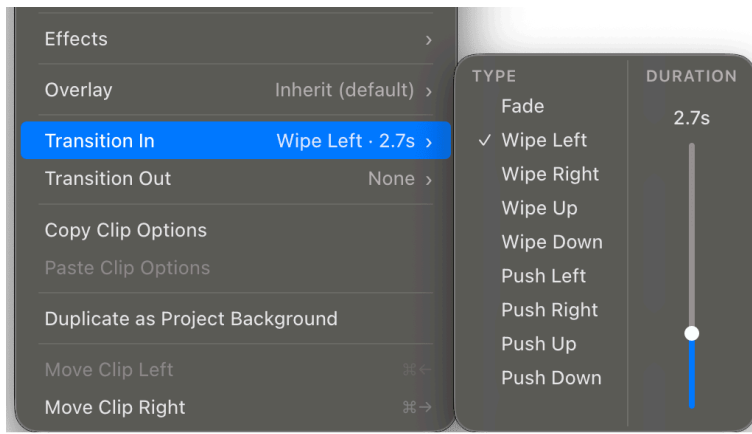
Rotate turns a Clip to any angle. Drag the slider anywhere from 0 to 360 degrees, or click the value and type an exact one. The slider snaps at 0, 90, 180 and 270 so the square turns are easy to hit, and any angle between them is available for a tilted or diagonal look.

Flip mirrors the Clip: **Normal**, **Horizontal**, **Vertical** or **Both**.

Both settings are live, so the original file is untouched. Neither applies to a Color Clip, since turning or mirroring a solid color changes nothing to see.

Transitions

Transition In and **Transition Out** decide how a Clip arrives on stage and how it leaves. Each one carries a shape and a duration together.



Each transition sets a shape and a duration in one place.

The shapes are:

- **Fade** dissolves between the two Clips.
- **Wipe Up, Wipe Down, Wipe Left** and **Wipe Right** reveal the incoming Clip across the frame, with the outgoing Clip staying put underneath.
- **Push Up, Push Down, Push Left** and **Push Right** slide the incoming Clip in and carry the outgoing one off with it.

Wipes and pushes are named for the direction the incoming Clip travels toward. Push Left brings the new Clip in from the right and carries the old one off to the left.

A duration of zero is a cut, whatever shape is selected, and the row reads **None**.

Transition In also sets the crossfade time between the previous Clip and this one, and it takes precedence over the previous Clip's Transition Out. If the Project has a Background Clip, a Clip transitions to and from that rather than to black.

For a video Clip with an audio track, the audio fades with the picture.

i NOTE

A Clip with a Transition Out keeps playing for the length of that transition after you stop it.

Velocity Sensitivity

Velocity Sensitivity lets a MIDI controller's playing dynamics reach the Clip's audio. Hit the pad harder and the Clip's audio plays louder.

The slider is per Clip, video and Audio Clips only, and set to 0 by default, so nothing changes until you ask for it. At 0 the Clip always plays at its Volume setting. Turn it up and a MIDI note's velocity scales that level, with the full response at 100%.

Volume stays the Clip's level in your mix. Velocity Sensitivity decides how far a single hit is allowed to move it.

Triggers that carry no velocity of their own play the Clip at full level whatever the setting: the mouse, the keyboard, a Stream Deck, OSC, the API and the Song Timeline all behave this way.

i NOTE

With **Retrigger Action** set to Stutter, Scratch, Reverse or Do Nothing, velocity applies only to the first hit. Every hit after that keeps the level the first one set, for as long as the Clip keeps playing. The Clip Editor Panel captions the slider **First hit only** when that is the case.

Where else these settings appear

- The [Clip Editor Panel](#) is the full set, with sliders and pickers.
- The [Clip menu](#) handles the Clip itself: duplicating, moving, deleting, and copying and pasting its options. **Edit Clip...** opens the panel where the settings above are set.
- [Launch Modes](#) covers what the trigger gesture itself does, which is a separate decision from any of the above.
- [Poly Mode](#) covers what happens when several Clips in one Song play at the same time.

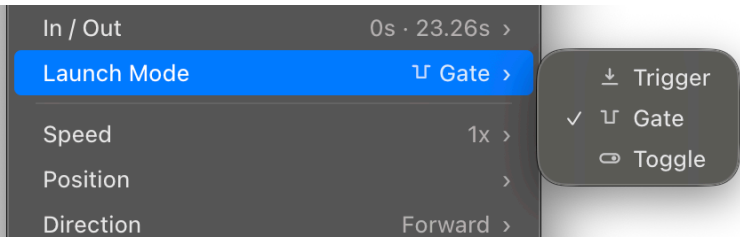
Launch Modes

Launch Mode decides what pressing and releasing a Clip’s trigger does. It is set per Clip, and it is the difference between a pad that fires a Clip, a pad that switches it on and off, and a pad that plays it only while you hold it down.

There are three modes:

Icon	Mode	Press	Release
⬇	Trigger	Starts the Clip.	Nothing.
⌵	Gate	Starts the Clip.	Stops it, over its Transition Out.
⌂	Toggle	Starts the Clip. Press again to stop.	Nothing.

Right-click a Clip and choose **Launch Mode** to set it.



Launch Mode is set per Clip in the Clip Editor Panel.

What each mode is for

Trigger is fire and forget. A press starts the Clip and nothing about the trigger stops it. It runs until its [End Action](#) setting says otherwise, or until you stop it, switch Songs, or start something that takes its place. A looping Trigger Clip keeps looping, which makes it the right choice for an ambient layer you want to leave running underneath everything else.

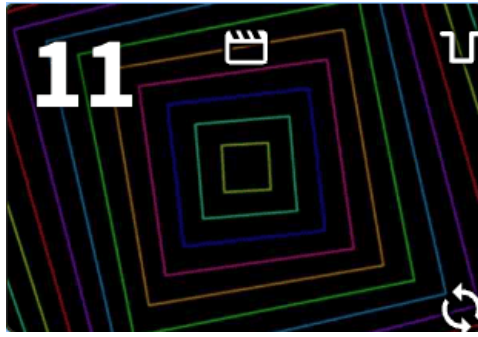
Gate is hold to play. The Clip plays while the pad is down and leaves when you let go, over its Transition Out. This is how you punch a stab in on a hit and take it straight back out, without a second gesture to end it.

Toggle is a switch. One press on, one press off. The second press stops the Clip over its Transition Out, whatever [Retrigger Action](#) says. Because Toggle always wins that argument, the Clip Editor Panel grays the Retrigger Action row for a Toggle Clip and shows Stop as the value in effect.

Launch Mode is a separate decision from Retrigger Action and End Action. Launch Mode is about the gesture, Retrigger Action is about a press that lands on a Clip already playing, and End Action is about the media running out.

The Launch Mode indicator

Every Clip tile in the Controller carries a small Launch Mode indicator, so you can read a Song’s behavior across the row without opening anything.



The indicator shows the mode the Clip will actually use.

The indicator always shows the mode **in effect**, which matters because a Clip you have never set follows its Song:

- In a Song set to Mono, an unset Clip is **Trigger**.
- In a Song with any other **Poly Mode**, an unset Clip is **Toggle**, so that clips you bring up you can also take down.

Set a Clip's Launch Mode yourself and that choice holds wherever the Clip goes. Hover a tile to see the mode written out in the tooltip alongside the Clip's name and type.

The Project's Background Clip has no Launch Mode. It is not triggered, so there is no gesture to interpret and no indicator on it.

What can release a Gate Clip

A Gate Clip needs a release, so it needs a controller that sends one. These all do:

- **MIDI** note-off, including a note-on at velocity 0 and a momentary CC returning to 0. The release is matched against the same note mapping as the note-on, so a pad mapped to a Clip gates it with no extra setup.
- **The number keys 1 through 9**. Hold the key, the Clip plays; let go, it stops.
- **The Song's Clip row in the Controller**. Press a tile and it plays, release and it stops. Releases are tracked per pointer, so on a touchscreen two fingers hold two Gate Clips at once.
- **The Preview area**, the same way as the Clip row.
- **Elgato Stream Deck**, on key up.
- **OSC**, the **command line** and the **API**, each of which has a release alongside its trigger.
- **Visibox Remote**, from its touch pads.

Send a release to a Trigger or Toggle Clip and nothing happens. Releases are ignored by every mode except Gate, so a controller can send them for everything and only the Gate Clips respond.

Gating from the Clip row

Because the Clip tiles fire on press rather than on click, a mouse or a touchscreen can gate a Clip directly from the Song's Clip row, with no controller involved.

That leaves dragging to reorder Clips without a gesture of its own, so hover a tile and a small grip appears. Drag from the grip to move the Clip; press anywhere else on the tile to play it.

If you would rather have the old behavior back, turn **Trigger Clips on Press** off in [Settings > General](#). Clips then fire on release, and the Preview area returns to normal dragging. A Gate Clip pressed from the Clip row or the Preview area behaves as a Trigger Clip, since there is no longer a release for it to act on. Gate Clips still work from MIDI, the number keys and every other release source above.

Setting Launch Mode across several Clips

Select several Clips and set Launch Mode once for all of them. **Copy Clip Options** and **Paste Clip Options** carry Launch Mode along with the rest of a Clip's settings, which is the quickest way to make a whole Song behave one way.

Where Launch Mode fits with Poly Mode

A Song set to Mono plays one Clip at a time, so a press starting a Clip is also what stops the last one. Launch Mode still applies: a Gate Clip in a Mono Song stops when you let go, and a Toggle Clip stops on its second press.

In a Song with a co-play Poly Mode, Launch Mode becomes the whole vocabulary. Gate Clips are the ones you hold in, Toggle Clips are the ones you leave up and take down later, and Trigger Clips are the layers you start and forget. See [Poly Mode](#).

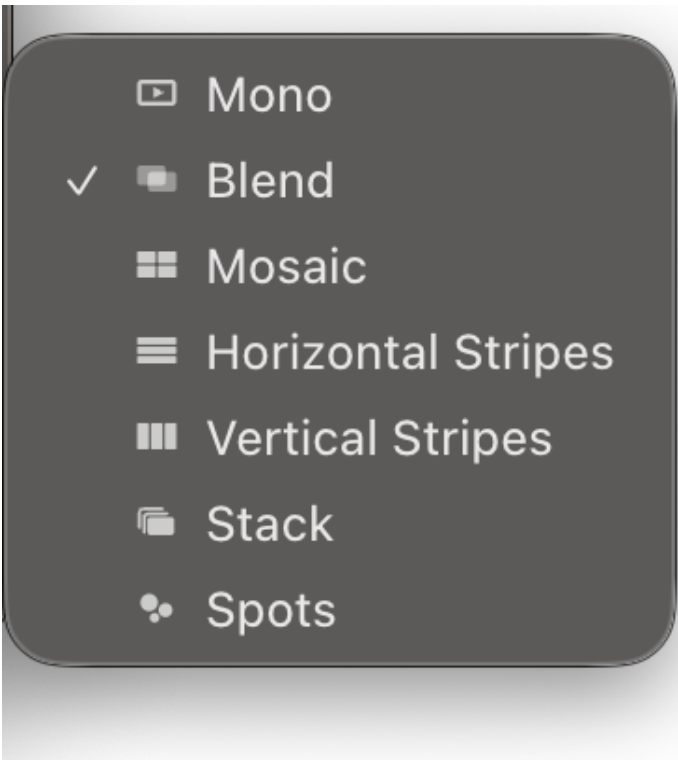
Poly Mode

Normally Visibox plays one Clip at a time: start a Clip and whatever was playing stops. **Polyoptic playback** lifts that limit. Set a Song’s **Poly Mode** and several of its Clips can hold the stage together, the way several notes held down on a keyboard sound together.

Poly Mode is set per Song, and it does two things at once: it turns co-play on, and it decides how the Clips that are playing divide the output.

Setting a Song’s Poly Mode

Right-click a Song and choose **Poly Mode**. You can also click the chevron on the Song row, which opens the same panel.



Poly Mode is set per Song.

The modes

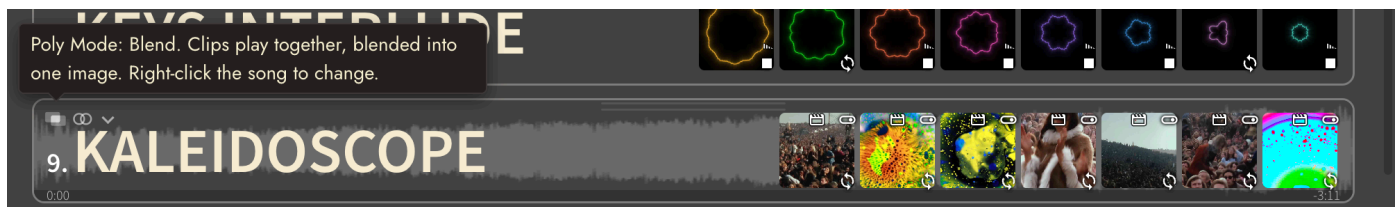
Mono (▶) is the default and the classic behavior. One Clip plays at a time, and starting a Clip stops the last one.

The other six let the Song’s Clips play together, and each lays them out differently.

Badge	Mode	What the output looks like
	Blend	The Clips are mixed into one image, sharing the brightness between them so the stage stays at a steady level however many are up.

Badge	Mode	What the output looks like
	Mosaic	Split screen. Each Clip gets its own tile, and the grid re-lays itself as Clips come and go.
	Horizontal Stripes	Full-width bands, one Clip per band, oldest on top.
	Vertical Stripes	Full-height columns, one Clip per column, oldest on the left.
	Stack	Every Clip fills the whole stage at full brightness, newest on top. Overlap comes from what each Clip is: its rotation, its Fit Mode, its transparency.
	Spots	Each Clip plays inside its own circle, placed where there is room and sized to fill it. A Clip that stops gives up its circle, so the next one to play lands somewhere new.

Each Clip keeps its own Transition In and Transition Out, so Clips arrive and leave smoothly rather than snapping into the layout.



Every Song row carries a badge for its Poly Mode. Hover it to read the mode.

Every Song row shows a badge for its Poly Mode, including Mono, so you can see at a glance which Songs in your set play more than one Clip at a time. Hover the badge for the mode by name and a one-line description.

Which Clips can play together

Only Clips in the **same Song** co-play. Triggering a Clip in a polyoptic Song still stops anything playing from a different Song, exactly as a Song change always has. So moving from one Song to the next behaves the way you expect, and co-play stays inside the Song you are on.

The Project's Background Clip is outside all of this. It plays underneath as always, and it never counts toward the limit below.

Starting and stopping the Clips

In a polyoptic Song, [Launch Mode](#) is what you use to bring Clips up and take them down.

- A Clip you have not set uses **Toggle** in a polyoptic Song: press to bring it up, press again to take it down.
- Set a Clip to **Gate** and it plays only while you hold its pad or key.
- Set a Clip to **Trigger** and a press never stops it. Trigger plus a looping end gives you a bed that stays up until you stop it another way.

Stop, stop-all and a Song change still stop everything, in any mode.

When the last Clip in a polyoptic Song stops, the Song's backing track keeps playing, so the next Clip you bring up lands on a track that never stopped. **Clip Stop Action** in the [Song Editor Panel](#) changes that.

Polyopsis: how many Clips at once

Polyopsis is the ceiling on how many Clips can play at the same time. It is one setting for the whole app, in [Settings > Output](#), and it can be set from 2 to 8. The default is 4.

Bring up one more Clip than Polyopsis allows and Visibox does not refuse it. It fades out the oldest Clip that is playing to make room, the way a synth drops its oldest note when you run out of voices. The Clip you triggered always plays.

Choose the number for the look you want rather than for the maximum. Mosaic at 4 gives you a clean 2 by 2, and the tiles get smaller and less even as the count climbs. Spots sizes its circles from this number too, so a lower Polyopsis gives you bigger spots.



TIP

Start at the default of 4 and turn it up only if you find yourself wanting more layers. A high ceiling with a busy Song can cover the stage in movement faster than it reads from the back of the room.

Where to start

Blend is the gentlest introduction: bring up two loops in the same Song and they mix into one image, no layout to think about. Mosaic is the one an audience notices, and it works well when the Clips are shot alike. Stack rewards Clips that have something to see through, so pair it with a Clip carrying transparency or a dark background. Spots is the loosest of them and turns a handful of short loops into a shifting collage.

Poly Mode changes nothing about a Song set to Mono. Every Clip setting on the [Clip Playback Options](#) page works the same way in a polyoptic Song as it does anywhere else.

Connecting Cameras & Video Inputs

Visibox's **Camera Clip** can display any live video feed, not only webcams. USB cameras are the most common input, but you can also feed in any HDMI source (a game console, a secondary computer, a video switcher, a broadcast camera) using an inexpensive HDMI capture adapter. If your computer sees it as a camera, Visibox can use it.

USB Cameras

USB cameras are usually called “webcams,” but hardware adapters and software let you connect many other digital cameras too. If a camera works with Zoom, it will probably work with Visibox. Most laptops also have a built-in camera: set one next to your drummer and add a Camera Clip for an instant drum cam.

HDMI Video Inputs

To pull an HDMI source into Visibox as a live feed (a game console, a video mixer, a second computer, a broadcast camera), you need an **HDMI capture adapter**. It is sometimes called a capture card or video capture device. These devices plug into your computer via USB and present any HDMI signal as a standard camera input, so Visibox (and Zoom, OBS, or any other app) can use it just like a webcam.

What you need:

- An HDMI capture adapter (USB devices from brands like Elgato or Magewell work well, as do generic ones. Look for “USB capture card” or “HDMI to USB capture”.)
- An HDMI cable from your video source to the capture adapter

To set it up:

1. Connect your HDMI source (game console, camera, switcher, etc.) to the capture adapter with an HDMI cable.
2. Plug the capture adapter into a USB port on your computer.
3. The device appears in Settings > Devices alongside your other cameras.
4. Add a Camera Clip to your Song and select the capture adapter as the source.



TIP

Most USB HDMI capture adapters need no drivers on macOS or Windows. They show up on their own as a standard video device. If your adapter isn't appearing, check that it's fully seated in the USB port and that no other app has exclusive access to the device.

NOTE

HDMI capture adapters vary in quality and latency. For live performance, look for adapters with low-latency pass-through and at least 1080p/30fps capture. Elgato Cam Link, Magewell USB Capture HDMI, and similar devices are popular choices in live production environments.

Virtual Cameras

Visibox also supports “virtual cameras”: software extensions that let your computer see an application as a camera. This software may capture a hardware camera input and add effects, or it may capture the output of a software application such as OBS or ManyCam. A hardware network camera that sends a video stream over the network may also create a virtual camera you can use in Visibox. Virtual cameras add camera or video feeds to your setup without saturating your USB bus.

NDI Cameras

NDI sends video over a network instead of a cable. An NDI sender on your network (a hardware NDI camera, another computer, or an app that broadcasts NDI) can be a Camera Clip too. NDI sources appear in the camera list alongside your physical inputs, with no capture hardware and no cable beyond the network.

By default Visibox connects to an NDI source only while its Clip is playing, which keeps network use down. Turn on **Connect to NDI Cameras Before Playing** in [Devices Settings](#) to hold the connection open from the moment the Project opens, so the first frame is instant.

Setting Up Cameras & Video Inputs

Every connected camera and video input shows up under [Settings > Devices](#). **Default Camera** picks the one a new Camera Clip uses.

Right-click a Camera Clip in the Controller window to point that one Clip at a specific device. Three cameras and an HDMI capture adapter can all run at once, each on its own Clip.

The menu bar offers the same list under Song > Add Camera Clip, which picks the device as it adds the Clip. When the computer has no camera or video input connected, that submenu reads **No Cameras Found**.

Resolution and frame rate

Each camera in Settings > Devices has its own **Capture Settings**: a **Resolution** and a **Frame Rate**.

Both start at **Auto**, which lets the camera pick. Set them yourself when you want a specific format, or to hold a camera down to something your machine can carry comfortably through a long show. The lists offer only what that camera reports it can do.

Under the two controls, a **Current:** line shows what the camera is actually delivering right now, as width by height and frames per second. If the camera is not running, it reads “not streaming”.

These are per camera and belong to the machine, not to the Project. The same Project on a different computer uses that computer's settings.

When a camera goes missing

Cameras get unplugged, and operating systems reassign device identifiers. When you open a Project on a machine where a Camera Clip's camera is not present, Visibox first tries to match it up by name and position in the device list. Most of the time the Clip repairs itself and you never notice.

When it cannot, the Clip shows **Unknown Camera**. To fix it, right-click the Clip and choose a camera from **Replace with Camera**.

If other Clips were using the same missing camera, Visibox asks whether to move those across too ("Update 3 other clips using this camera?"). Say yes and the whole set is repaired in one step, which counts as a single undo.

NOTE

Two identical cameras report identical names to the operating system. Visibox falls back to their position in the device list, which follows the port rather than the unit. Swap two matching cameras between ports and their Clips swap with them.

Audio in Visibox

Audio Outputs

Every connected audio interface shows up under Settings > Audio > Output Device. The first item in that list follows the output you have chosen in your system settings, so Visibox works out of the box with your computer's built-in output.

If you have a better interface for the show, point Visibox at it and leave your system output on the built-in speakers. Alerts and notification sounds then stay on your laptop instead of going out to the audience.

The full control reference is on [Audio Settings](#).

Mute

Visibox has a true mute control that silences all audio output while preserving your volume setting. When you mute, the volume slider stays in position, so you can see exactly where it will return to when you unmute. Moving the volume slider while muted unmutes.

You can toggle mute from the Control Bar, from Settings > Audio, or with a keyboard shortcut. Turn on Remember mute state in Settings > Audio if you want Visibox to start in the state you left it.

Multichannel Audio

Most built-in outputs are stereo, but Visibox also plays across multichannel interfaces.

On the Mac, Visibox sees every output your interface offers, with no extra setup.

On Windows there are two paths. The default path goes through the Windows audio system and reaches up to 8 outputs, after you configure the interface in Windows Sound. The ASIO path talks to your interface's own driver and reaches the channel count the interface actually has. Both are described below.

Most multichannel interfaces ship with configuration software that sets the output routing. You might route your interface's ADAT outputs as channels 1 to 8, for example. Visibox follows that routing and spans your multichannel audio across the outputs starting at channel 1.

Channel assignment comes from the source file itself. Visibox sends the file's first channel to output 1, its second to output 2, and so on. There is no per-clip routing control, so the order of the channels in the file is the order they reach your interface.

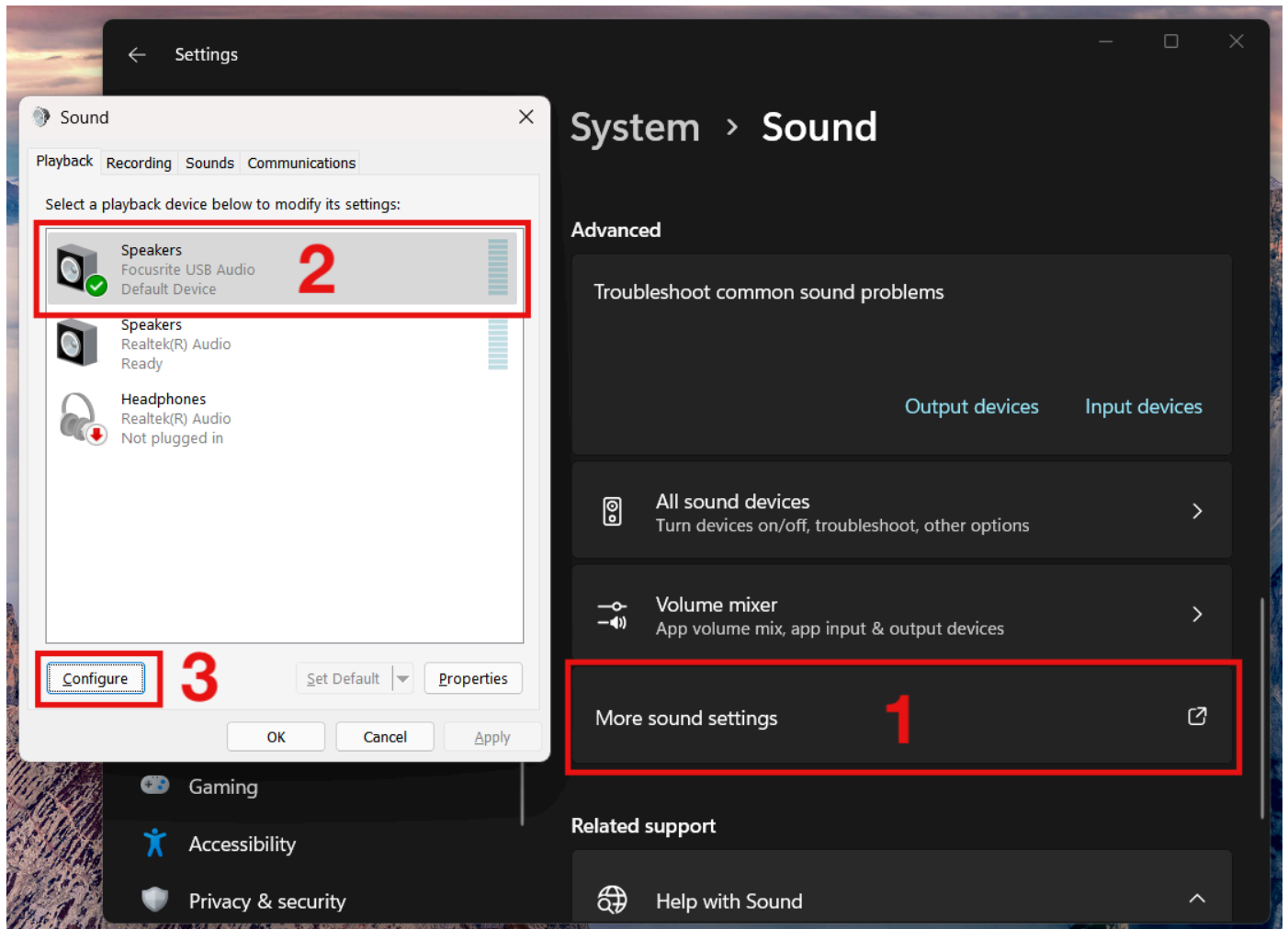
Configuration on Mac

Visibox detects connected multichannel interfaces on its own. Open Settings > Audio, choose your interface under Output Device, and all of its outputs are available.

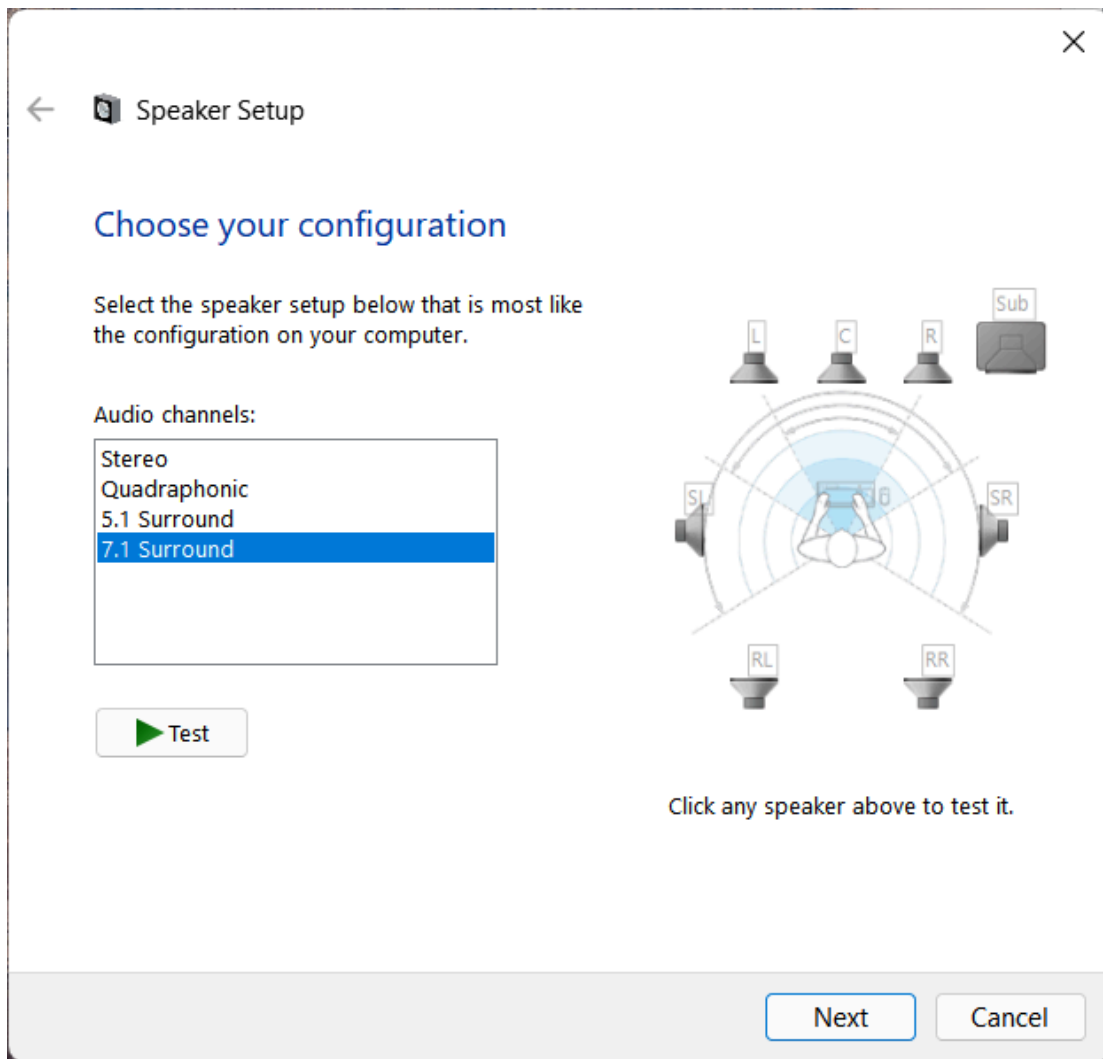
Configuration on Windows

Once the interface is connected and its drivers are installed, configure it in the Windows Sound settings:

1. Open Settings, then go to System > Sound > More Sound Settings.
2. Select your audio interface and click the Configure button.
3. In the window that opens, choose 7.1 Surround Sound to use all 8 outputs.



Open the Sound panel. Then click “More sound settings” to open the window. Select your multichannel audio interface. Then click the “Configure” button to set it up for multichannel output.



In the popup window, select “7.1 Surround” to enable all 8 outputs.

If your interface is connected and you do not see it listed, or you do not see a 7.1 Surround option, check the following:

- You have the latest drivers and system updates.
- Everything installed correctly.
- The interface is set up for multichannel output in its own configuration software.

ASIO Output on Windows

Windows Sound tops out at 8 channels, which is short of what a 16 output interface can do. Visibox can drive those interfaces through ASIO instead, talking to the interface’s own driver and using its full channel count. There’s no Windows Sound configuration to do. Turn on Settings > General > Enable Beta Features to see this.

Pick an ASIO device from the Control Bar. Click the output level meter at the right of the Control Bar to open the menu. It lists every output device Visibox can use, with your ASIO interfaces added below the operating system’s own devices. Each ASIO entry names the interface and its channel count, and reads “ASIO - Your Interface - 16ch (Beta)”. ASIO interfaces do not appear in the Output Device list in Settings > Audio.

How many channels Visibox sends depends on the Channels setting:

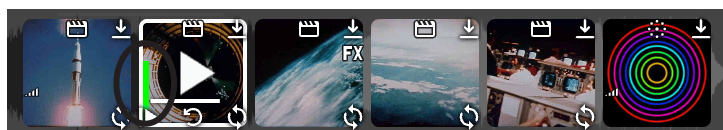
- **Multitrack** sends discrete channels spread across the width of the interface.
- **Stereo** and **Mono** fold everything down to two channels or one, the same as they do on any other device.

Choosing an ASIO device silences the computer's own output, so only the interface plays. Choose another device, or unplug the interface, and the computer's output comes back.

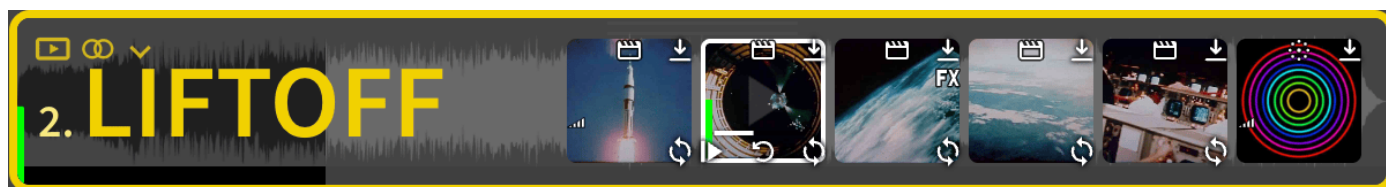
Clip & Song Audio Metering

Visibox displays real-time audio level meters on individual Clips and Songs in the Controller window. A green vertical bar appears on the left edge of any Clip or Song that is currently producing audio. That makes it easy to see at a glance where your audio is coming from. It might be a video Clip with embedded audio, an audio Clip, or a backing track attached at the Song level.

The meters respond instantly to audio peaks and fade smoothly as levels drop. If the audio level clips, the meter turns red to warn you that distortion may be occurring. Muted Clips and Songs do not display a meter.



A playing Clip carries its own level meter.



Clip and Song meters together, so you can see which one is making the sound.

TIP

During a performance, the Clip and Song meters are a quick way to diagnose audio issues. If you hear unexpected audio, glance at the Controller and the meters will show you exactly which Clip or Song is producing sound.

System Audio Input

Visibox can capture audio from other applications running on your computer and use it as input for [Visualizers](#) and audio-reactive [Effects](#). Visualizers and Effects like Bass Pulse, Audio Shake, Spectrum Bars, and Disco Audio Flash can then react to music playing in Spotify, a web browser, a DAW, or any other app producing sound. They react to audio inside Visibox too.

Turn it on in Settings > Audio > Enable System Audio Input.

NOTE

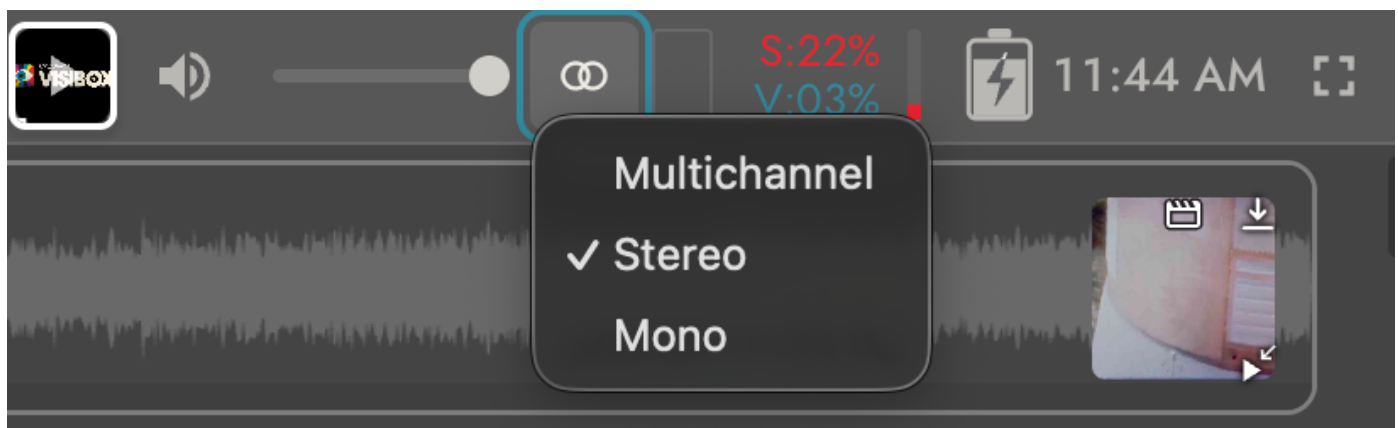
On macOS this needs a system permission. On macOS 14.2 and later, the system asks you for System Audio Recording permission the first time Visibox captures, and no restart is needed. On earlier versions, grant Visibox Screen & System Audio Recording under System Settings > Privacy & Security, then restart Visibox.

Downmixing

Visibox offers three modes for audio output: Stereo, Mono, and Multichannel. Choose one in Settings > Audio > Channels, where the third mode is labeled Multitrack. You can also click the audio mode mark next to the output meter in the Control Bar at the top of the Controller window.

The modes work as follows:

-  **Mono:** Audio plays in dual mono, with the same signal sent to channels 1 and 2. Stereo source audio is downmixed to mono, and multichannel source audio is downmixed to mono with all channels mixed together.
-  **Stereo:** Audio is output in stereo, using channels 1 and 2 of your interface. Mono source audio plays equally in both channels. Multichannel source audio is downmixed to stereo with odd channels on the left and even channels on the right.
-  **Multichannel:** Audio is output in its original multichannel format, with each channel sent to its corresponding output. Nothing is downmixed. Mono files play in dual mono on channels 1 and 2. Stereo files play in stereo on channels 1 and 2. Multichannel files play across all available outputs, starting at channel 1, and any channels beyond your interface's width are not output.



Click the mode mark next to the output meter in the Control Bar to change the audio output mode.

Multichannel Metering

When you have multichannel audio files (more than 2 channels) in your Project and Visibox is in multichannel playback mode, Visibox displays a multichannel meter in the Control Bar. The meter shows playback levels for any playing audio or video file. If your audio interface has fewer output channels than the

file, the extra channels are drawn as outlined meters. A 6-channel audio file on a 4-channel interface shows channels 5 and 6 outlined.

Meter bars run green through the working range, turn amber as they approach full scale, and go red at full scale.

Bars render at their full width through 8 channels and compress from there, so a stereo mix reads clearly rather than as a pair of slivers. The meter itself is capped in width, so a high channel count leaves the clock and battery readouts where they are.



An 8 channel audio file playing on a stereo interface. Channels 3-8 are outlined because they are not being output. Note the multichannel mark next to the output meter in the Control Bar.



An 8 channel audio file playing on a 32-channel interface.

If Your Interface Disconnects Mid-Show

Audio keeps playing. If the device Visibox is using goes away, whether someone kicks a USB cable out or a dock drops off, Visibox moves the audio to another device rather than falling silent. It chooses in this order:

1. Your chosen ASIO device, if it is present.
2. Your chosen output device, if it is connected.
3. Whatever is already playing, if it is still connected.
4. The first available device.

A notification tells you what happened, naming the device that went away and the one now playing. Plug the interface back in and audio returns to it, with a second notification confirming the reconnection.

Step 3 is the one that matters on show day. Connecting an unrelated device while your interface is missing leaves the output where it is, so the sound stays put instead of jumping to whatever was plugged in last.

Combining Stem Files

Visibox combines multiple audio files into a single multichannel file for you, up to 31 channels. In addition to playing multiple Clips per Song, you can attach a single audio file to each Song: mono, stereo, or multichannel, such as WAV, AAC, MP3, or FLAC.

Outputting a multichannel file from a DAW (Logic, Pro Tools, and the like) is far from straightforward. Most of these tools are built for “surround” audio, and that workflow gets in the way when what you want is one file holding several stems: drums, bass, guitar, vocals. Combine them in Visibox instead. Here’s how:

1. **Output your stems** from your DAW as separate audio files, uncompressed (WAV) or compressed (AAC, MP3). Four stereo stems, say a drummer’s click, a band click, keyboards, and backing vocals, combine into one 8-channel file.

2. Name the files so they **sort alphabetically** in the order you want them to play: “01 Click Track Drums”, “02 Click Track Band”, “03 Keyboards”, “04 Backing Vocals”.
3. In Visibox, create or **find your Song and attach the audio files** to it, all at once. You can do this by either:
 - a. Right-clicking on the Song and selecting Attach Audio, then selecting the multiple stem files you want to combine.
 - b. Dragging all of the files from your file browser directly onto the Song in the Controller window.
 - c. Copying the files from your file browser, then pasting them onto the Song in the Controller window.
4. Visibox asks you to **confirm the combination of the files**. Confirm the combine and Visibox generates a new multichannel audio file that holds every stem in the order you specified. When you drag or paste the files, the dialog also offers **Keep Separate**, which adds each file as its own Clip instead.

The finished file attaches to the Song and plays like any other audio file. An icon with three interlocking circles () marks the Song as carrying multichannel audio.

Watch this video: <https://youtu.be/PrI8UXcIq5Y>

Merging audio stems in Visibox. [YouTube link](https://youtu.be/PrI8UXcIq5Y)

WARNING

Channel limit

31 channels is the ceiling, whether the audio sits in an audio file or in a video file. If the files you select add up to more than that, Visibox tells you the total and cancels the combine rather than making a file it cannot play. A single file that is already wider than 31 channels is turned away when you import it, for the same reason.

Combining Stem Files Into a Video File

You can also combine audio stems into a video file. This keeps a video, such as a music video with separate stems for drums, bass, and backing vocals, in one file with its audio. Follow the same steps as above, and include a video file among the files you add to the Song. Visibox recognizes the mix of audio and video and asks you to confirm the combine.

Existing Multichannel Audio and Video Files

If you already have a multichannel audio or video file that you want to use in Visibox, drag it onto the Song in the Controller window. Visibox recognizes it as a multichannel file and plays it back across all available outputs. Keep in mind that the layout of the channels in, for instance, a 5.1 surround sound file may not match the layout of your audio interface. Visibox plays the channels in the order they are defined in the file, so you may need to adjust your audio interface’s routing to match.

Output, Canvas, and Displays

Visibox paints one picture and then hands out pieces of it. That picture is the output canvas. A projector, a second projector, an NDI stream, a DeckLink card and a screen recorder can each take the whole canvas or a rectangle of it, at the same time.

This page covers what the canvas is and how to send it places. The controls themselves live in two Settings tabs: [Output Settings](#) for the shape of the picture and [Canvas Settings](#) for where it goes.

The output canvas

Each destination you turn on is an output target. A target is a physical display, an NDI output, a DeckLink output, Syphon or Spout, or the recording capture region.

Every enabled target holds a rectangle on a single shared canvas, laid out in the arranger in Settings > Canvas. The canvas is the bounding box of all those rectangles. Visibox draws the whole canvas once, and each target shows the slice its rectangle covers.

Two rectangles on the same spot show the same content. Two rectangles side by side show two halves of a wider picture. That is the whole model.

The canvas cares about a screen's size, not its pixel count. A high-density (Retina or HiDPI) screen takes up the same room on the canvas as a standard screen of the same physical size.

Sending output to more than one display

1. Connect the displays and open Settings > Canvas. Window > Arrange Displays... opens the same tab.
2. Displays that are not carrying output sit in the Unused tray. Drag one onto the canvas to turn it on.
3. Drag its rectangle to say what part of the picture it shows.
4. Select a rectangle to open the Transform panel and set Position, Width and Height, Scale, Rotation and flip.

Click **Identify Displays** at any time to flash a number on every connected screen.

One display can be the Controller display, and one can be the Lyrics display. A display doing either of those jobs is not an output, and no display can do two of the three at once. Set which is which from the display's **Role** in [Canvas Settings](#).

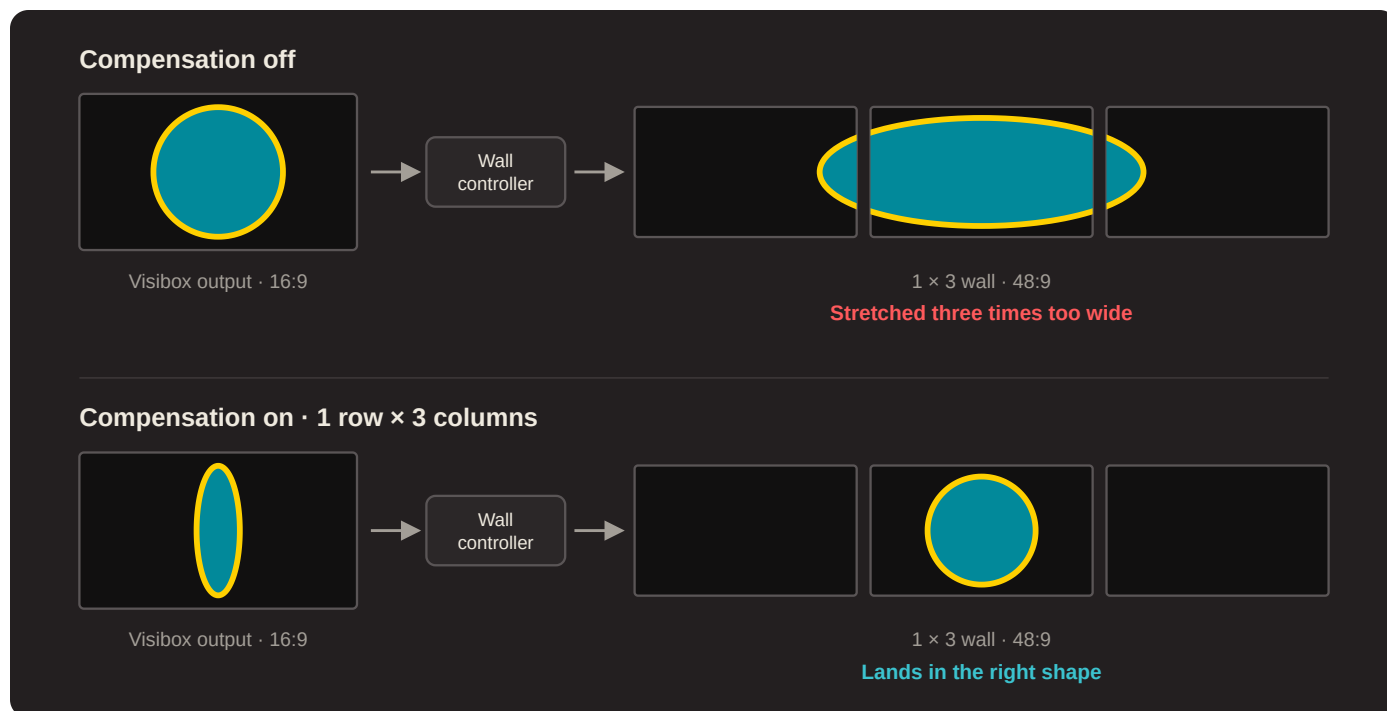
Video walls

A hardware video wall controller takes one signal and splits it across a grid of panels, then shows each panel its tile full screen. When the grid is not square, the wall's overall shape does not match the signal's, so the picture arrives stretched. A 1×3 wall spreads a 16:9 frame across a 48:9 surface, and everything on it is three times too wide.

Video Wall Controller Compensation pre-distorts the feed by the inverse amount so it lands correctly. Select the display feeding the controller in Settings > Canvas, then set Rows and Columns to match the panel grid. Two projectors side by side is 1 row by 2 columns. Four televisions stacked is 4 rows by 1 column. Rows and columns each go from 1 to 16, and 1 × 1 means no wall. The compensation depends only on the ratio between rows and columns, so the panel count costs nothing.

Rows and columns each go from 1 to 16, and 1 × 1 means no wall. The compensation depends only on the ratio between rows and columns, so the panel count costs nothing.

The setting is per display. A rig with one walled projector and one plain confidence monitor gets compensation on the projector only.



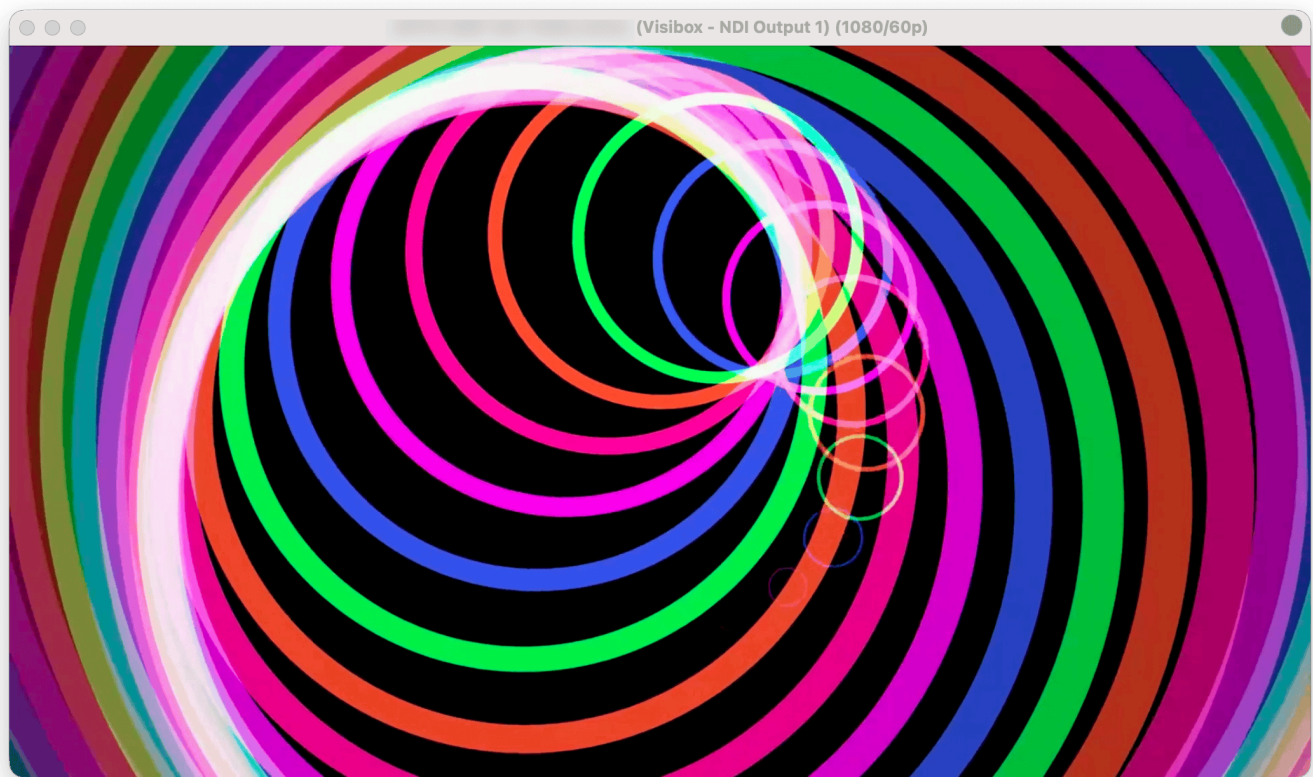
A 1 × 3 wall. With compensation off, the controller stretches the picture three times too wide. With it on, Visibox squeezes the output first so it lands in the right shape.

Compensation assumes the panels have the same shape as the signal feeding them, which covers a 16:9 output driving 16:9 panels. A wall built from panels of a different shape needs the wall controller's own correction instead.

NDI output

NDI sends video over the network, so a receiving machine on the same network can take the picture without a cable. Add one with **Add NDI Output** in Settings > Canvas. Turn on Settings > General > Enable Beta Features to see this.

Each NDI output behaves like any other target. It gets a rectangle on the canvas, so you choose whether it carries the whole picture or one slice of it. In the Transform panel you can rename it, set its Width and Height, and remove it. The name is what receivers see in their source list. The Output master audio mix travels with the video.



A Visibox NDI output as it appears to a receiver on the same network.

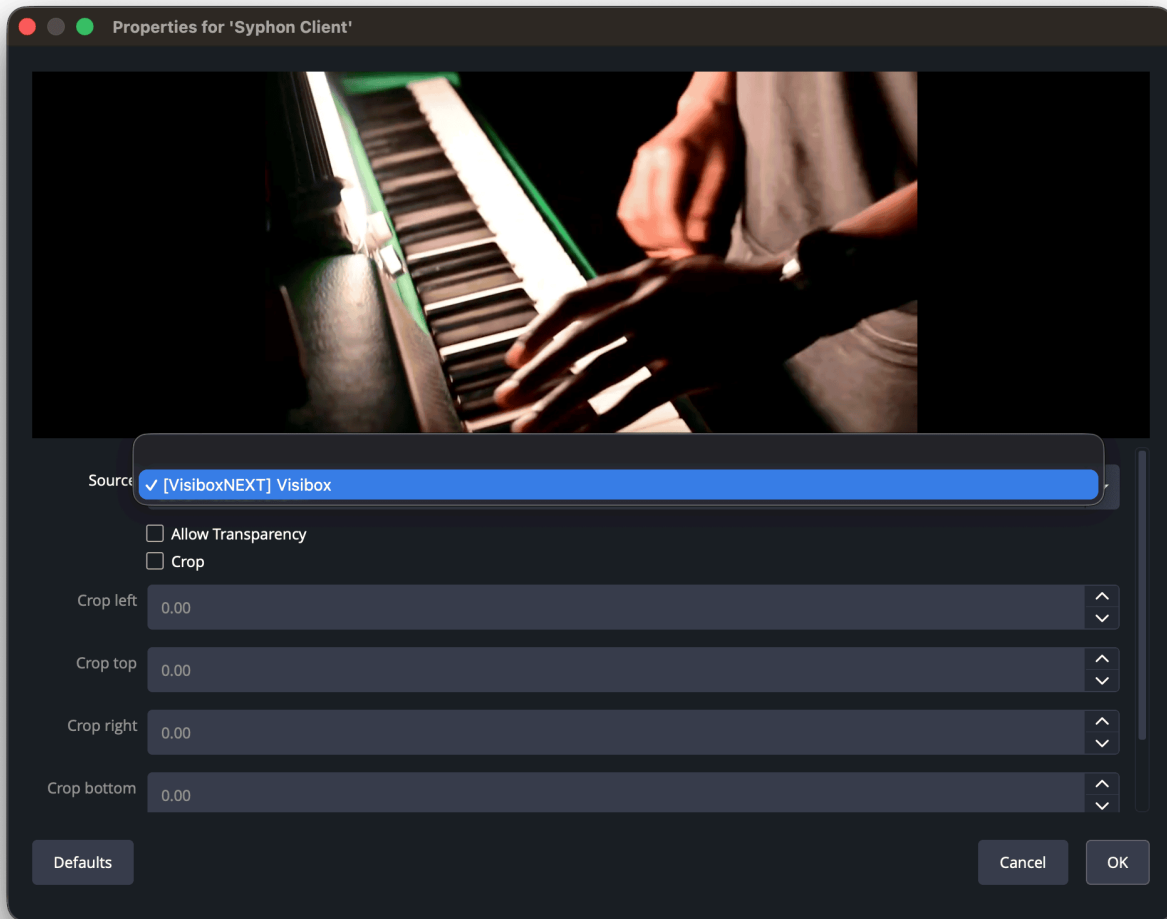
DeckLink output (SDI)

A DeckLink output plays out over SDI through a Blackmagic DeckLink card. **Add DeckLink Output** in Settings > Canvas needs Blackmagic Desktop Video installed and a DeckLink device connected. With more than one device connected, a picker beside the button chooses which card the new output plays out on. Turn on Settings > General > Enable Beta Features to see this.

A DeckLink output takes a rectangle on the canvas like any other target. Select it to set its **Video Standard**, the device's own display mode. New outputs start on the card's first 1080p mode. The panel names the device the output belongs to, and says so when that device is not currently connected. The Output master audio mix is carried on the SDI stream.

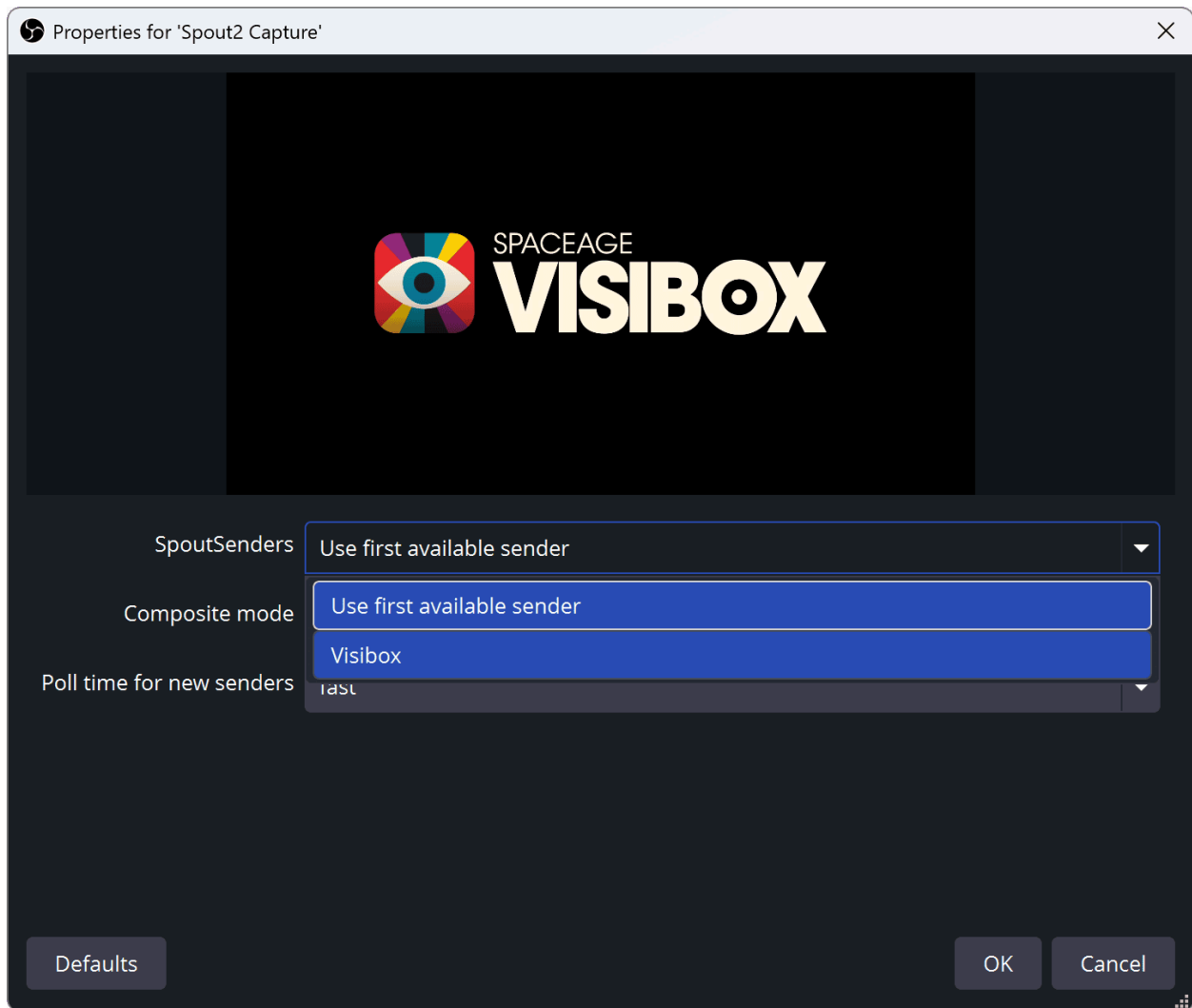
Syphon and Spout

Syphon Output in Settings > Output shares the live output as a Syphon source, so other macOS applications such as OBS and Resolume can receive it with no cable. Turn on Settings > General > Enable Beta Features to see this.



A Visibox Syphon share as it appears to another application on the same Mac.

Spout Output in Settings > Output shares the live output as a Spout source, so other Windows applications such as OBS and Resolume can receive it with no cable. Turn on Settings > General > Enable Beta Features to see this.



A Visibox Spout share as it appears to another application on the same PC.

Once the switch is on, the share also appears as its own rectangle in the arranger. Place the rectangle over part of the canvas to hand over a slice, or set its [Mirror](#) to Full canvas to hand over the whole show whatever shape the rectangle is.

The shape of the picture

Rotation, flip, an enforced content aspect ratio and rounded corners are in [Output Settings](#). They shape the picture itself, before any of the targets on this page get hold of it, rather than dressing up one destination. Rotation and flip wait for full screen. **Only Rotate/Flip/Stretch in Full Screen** is on by default, so you will not see either of them in a window until you turn it off.

Output Window Size and **Output Window Shape** sit on the same tab and are the exception. They set how large the Output window is on screen and change nothing about the picture. That is why a small window no longer starves a recording or a stream of the pixels it asked for.

Undoing output changes

Anything you change on the Output and Canvas tabs can be taken back. Press **⌘Z** while the tab is showing, or use **Edit > Undo**, and the Edit menu names the step it is about to reverse. Each tab keeps its own steps, and they last only while the Settings window is open.

Recording the output

Visibox can record what the audience sees, with the master audio mix, to a video file.



The Record button sits in the Control Bar at the top of the Controller window.

Click **Record** in the Control Bar and choose:

- **Start Recording Now** begins immediately.
- **Start Recording on Next Trigger** waits, then begins the moment the next Clip starts playing. Use this to start a take on the downbeat instead of a few seconds before it.

While armed, clicking Record again cancels. While recording, clicking Record again stops. The Play menu's Recording section offers whichever of those applies right now, always on **⌘R**.

When you stop, Visibox asks where to save. The default location is your Videos folder. Files are H.264 in an MP4 container where the machine supports it, and VP8 in a WebM container otherwise. If you cancel the save dialog the take is not thrown away: it stays in a temporary file, and Visibox logs where.

By default a recording captures the whole canvas. To record part of it instead, turn on the **Recording Capture Region** in [Settings > Canvas](#) and place its rectangle over the part you want.

The region starts turned off, so the first time you record, Visibox asks **Record the whole canvas?** and tells you the size it would use. You can:

- **Record Full Canvas** and get on with the take.
- **Set Up Capture Region...**, which cancels this recording and opens the arranger with the region already switched on and snapped to the canvas, ready to place.
- **Cancel**, and nothing is recorded.

The prompt carries a **Don't ask again** box, which applies only to **Record Full Canvas**. To bring the prompt back, turn on **Ask before recording without a capture region** in [Settings > Canvas](#).

With a region set up, **Recording Size** on the Transform panel picks the resolution of the file, and **Match Canvas** snaps the region to cover the whole canvas.

Optimizing Visibox

This page is organized by symptom. Find what is going wrong and work down its list, cheapest fix first.

For the show-day routine rather than tuning, see [Setting Up for Live Performance](#).

Playback stutters

Work down this list in order. The first two cost nothing and often finish the job.

1. **Plug in.** A laptop on battery throttles its CPU and GPU. This is the single most common cause, and the fix takes a second.
2. **Quit everything else**, especially a browser with many tabs, a DAW, and anything syncing files. Visibox is competing with them for the same GPU and the same disk.
3. **Lower the Resolution Limit** if you are driving more than one screen. Visibox renders the show once into a single picture that every display and every output samples from, and the more pixels it carries the more work each frame is. Setting the limit to 1080p costs some sharpness and usually gives the frame rate straight back.
4. **Optimize your media.** The Media window tells you which files are heavier than they need to be and converts them. This is the fix that lasts, and it is the slowest to apply, which is why it is fourth.
5. **Move the Project to a fast local drive.** See [Storage](#) below.



TIP

Try the Resolution Limit before you re-encode anything. It is one click, it is reversible, and on a multi-display rig it usually helps more than anything you can do to the files.

Playback is smooth, but the picture is soft

You may have the Resolution Limit set lower than the screens need. Put it back to **Automatic** in [Settings > Canvas](#) and see whether the frame rate holds.

On a rig that mixes a high-resolution laptop screen with a standard projector, [Sharp rendering](#) gives the high-resolution screen its full detail, at the cost of more work per frame. It does nothing at all when every display has the same pixel density.

It was fine yesterday and it is sluggish today

Restart Visibox, then restart the computer.

This sounds like a joke and it is not. If the graphics system crashes enough times, the operating system stops using the graphics card and falls back to software for the rest of the session, which is much slower. Visibox warns you when it detects this, but if you missed the warning, a restart is the fix.

Adding media is slow

Real-time antivirus reads every file Visibox writes, which means every large video as it imports and every automatic save. Excluding the Project folder from scanning removes that delay while leaving the rest of your computer protected.

Visibox offers to do this the first time you open a Project. If you dismissed the offer, see [Antivirus and file indexing](#).

Visibox marks its media and backup folders so Spotlight skips them, which it does on its own with no setting to find. See [Antivirus and file indexing](#).

If you run third-party antivirus, exclude the Project folder in that product.

Importing also copies each file into the Project, so the speed of the drive you are importing from and the drive the Project lives on both matter.

Storage

Keep the Project on a fast local drive. An internal SSD is best. An external SSD over USB-C or Thunderbolt is fine.

Do not run a show from a network drive or a cloud folder. Dropbox, iCloud Drive, OneDrive, and Google Drive all sync in the background and hold files open while they do it. Any of them can replace a local file with a placeholder that has to be downloaded before it will play, which is a stutter at exactly the wrong moment. That is why Visibox tells you to copy a Project to a local drive when it finds one on a network volume.

Use cloud storage to move a Project between machines. Copy it off before you perform.

Leave room. Visibox writes as it goes: the Project on every change, a backup every few minutes, thumbnails, and any converted media. A full disk means changes stop being saved, and Visibox tells you when that happens. Do not let a show-day disk sit near full.

Keeping the Project itself lean

- **Run Clean Up Unused Files...** now and then. Visibox keeps a copy of replaced media so undo works, and this is how you get that space back. It cannot be undone, so do it when you are between shows rather than during one.
- **Trim rather than pre-cut.** [In and out points](#) play a section of a file, so one long file can serve several Clips without making several copies of it.
- **Duplicate rather than re-import.** Duplicating a Clip points at the same media file, so it costs no extra space. Importing the same file twice does not either, because Visibox recognizes a file it already has and reuses it.

What Visibox already does for you

Some of the work is done before you ask. Knowing it is happening explains a few things that would otherwise look odd.

- **Media is prepared in the background.** Playing a Clip backward, or bouncing it, or crossfading its loop, needs a prepared copy of the media. Visibox makes it in the background and the Clip plays normally in the meantime, then picks up the prepared version when it is ready.
- **Files are pre-read on open.** Visibox reads the start of every media file when a Project opens, so the first time you fire a Clip it is not also waiting on the disk, the antivirus, or a cloud download.
- **Nothing heavy runs during a show.** Background conversions hold off while anything is playing and resume when the output goes idle.
- **Update checks stop** while the Output, Controller or Lyrics window is full screen.
- **Audio finds another route.** If the audio device you chose disappears, Visibox moves the sound to another one and tells you where it went, rather than going silent.

When to suspect the machine rather than the Project

Open the Demo Project from **Help > Download/Open Demo Project...** and play a Clip from it. If that stutters too, the problem is the computer or its setup rather than anything in your Project, and the list at the top of this page is where to start.

Setting Up for Live Performance

You arrive at the venue, you have twenty minutes, and the projector belongs to someone else. This page is the show-day routine: get the picture on the screen, get it aimed and focused, and check the things that fail quietly.

For rigs with more than one screen, a video wall controller, NDI, a DeckLink card or a screen recording, see [Output, Canvas, and Displays](#).

Basic hardware setup

Visibox is built to run on any current Mac or Windows laptop connected to almost any projector or screen. A faster machine with a solid-state drive loads and plays high-resolution media better. A bright projector (3000 lumens or more) at 720p or better holds up in a room with stage lighting.

Use two displays

You can run Visibox on one screen by putting the Output window into full screen, but that hides the Controller and leaves you flying blind: no song list, no preview, no clock. Two displays is the normal setup. The Controller stays on the laptop screen, and the Output goes to the projector.

Set which display carries which window in Settings > Canvas. See [Output, Canvas, and Displays](#).

Common laptop video outputs

- USB-C (or Thunderbolt 3 and later)
- HDMI
- DisplayPort or Mini DisplayPort
- DVI
- VGA
- Thunderbolt 1 or 2

Common projector inputs

- HDMI
- VGA
- DVI

Adapters exist between HDMI and all of these, so HDMI is usually the common denominator. A new MacBook Air and an old VGA-only projector need a USB-C to HDMI adapter, a long HDMI cable, and an HDMI to VGA adapter. Adapters vary a lot in quality, so carry a spare and try another if the first one gives you nothing.

Going full screen

Choose **Window > Present Full Screen** ( **F**) to put the Output window full screen on its display. Press  to come back out.

If the Output and the Controller have ended up on the wrong screens, **Window > Swap Presentation Displays** ( **X**) trades them.

TIP

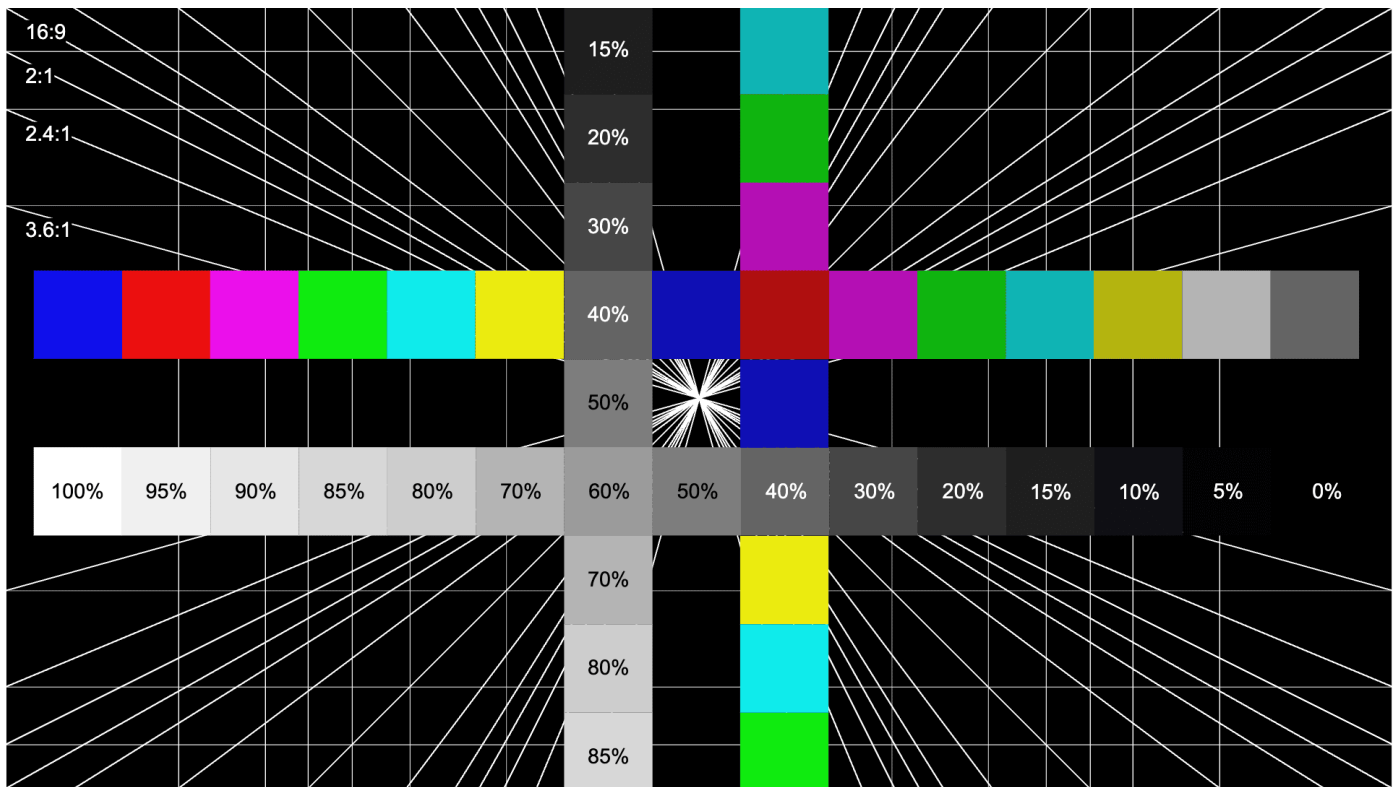
Use full screen rather than dragging the Output window's corner out to cover the screen. A hand-resized window keeps its title bar and its edges, and both of those end up on the projector.

NOTE

On macOS, full screen across two displays needs the “Displays have separate Spaces” setting turned on in System Settings > Mission Control. Visibox tells you if it is off and offers to fix it automatically. The change takes effect after you log out and back in, so check this before show day rather than at soundcheck.

Aiming and focusing with Calibration and Focus

You often have minutes to aim, focus, and calibrate a borrowed projector. **Window > Calibration and Focus** ( **K**) throws up a test image built for exactly that: fine lines for focus, saturated colors for color calibration, grayscale steps for contrast, and marked aspect ratios so you can tell what shape the screen actually is.



The Calibration and Focus image tells you the projector's aspect ratio and gives you something sharp to focus on.

Press  again to take it away.

Fitting the picture to the screen

A borrowed projector may be ceiling mounted, behind the screen, or a different shape from your content. Visibox has a control for each of those, and they are all in Output Settings:

- **Rotation** for a projector hung upside down, or for a vertical banner screen.
- **Flip** for rear projection or a ceiling mount.
- **Content Aspect Ratio** to hold your content at a fixed shape no matter what shape the screen is.
- **Round Corners** to soften the output's edges.

All four, and the switch that limits them to full screen only, are covered in [Output Settings](#). That switch is on by default, so check the result full screen on the display you are correcting for, not in the windowed Output.

NOTE

Resolution

A video's resolution is how many pixels wide and high it is. Displays have resolutions too: a typical 720p projector is 1280 by 720.

A low-resolution video shown on a higher-resolution display gets scaled up and can look soft or blocky. A much higher-resolution video looks great but is harder to play back smoothly. Where you have the choice, pick media in the same ballpark as the screen you are playing it on.

When playback stutters on several displays

Several large screens can ask for more pixels per frame than a modest machine can render smoothly.

Resolution Limit in [Settings > Canvas](#) caps the size of the picture Visibox renders, however many screens are attached, and gives the frame rate back at the cost of some sharpness.

It takes one click and it is reversible, so try it before you re-encode any media. [Optimizing Visibox](#) explains how it works and where it sits in the tuning order.

Soundcheck checklist

- Plug the laptop into AC power. Do not run a show on battery.
- Connect the projector and confirm you have a picture.
- Run Calibration and Focus () and check the aim and focus.
- Put the Output into full screen () and confirm it landed on the right display.
- If a MIDI controller drives the show, trigger a few Clips and check that any indicator lights respond.
- If the Project uses sound, play a Clip with audio and confirm it reaches the desk and the monitors.

Using MIDI in Visibox

You can run a whole show from the mouse and the computer keyboard. Visibox comes alive when you plug in a MIDI controller instead. Your hands stay on the pads, and a controller with lights tells you what is happening without you looking at a screen.

What MIDI can drive

Anything on the MIDI Map, which covers most of what you do during a show:

- Trigger a Clip by its position in the Song.
- Play, Stop, Pause and Resume.
- Move between Songs and between Clips, with or without stopping playback.
- Ride the master output volume from a fader.
- Sweep a Real-Time Control, which in turn moves a Clip or Effect parameter.
- Jump straight to a Song, either by scrubbing the Song list with one knob or by program change.

The assignments themselves live in [Settings > MIDI](#), which is also where you arm a row and press the pad you want it to answer to. Changes to the map can be taken back while that tab is showing: press  or use **Edit > Undo**, and the Edit menu reads **Undo Edit MIDI Map**. The steps last only while the Settings window is open.

Attaching and configuring MIDI devices

Most modern MIDI devices connect over USB. Plug the device into your computer, directly or through a hub, and Visibox picks it up. If your device only has DIN sockets, an inexpensive MIDI-to-USB interface bridges it.

Visibox listens for MIDI notes and continuous controller (CC) messages. That covers nearly every keyboard, pad grid and drum machine on the market. A few MIDI foot controllers send only program change. Visibox can use that to select Songs, but not to trigger Clips, so check for note or CC output before you buy one for this job.

New devices are enabled as inputs automatically. To stop Visibox listening to one, pick it under **Input** in [Settings > MIDI](#) and uncheck **Enabled**. That is worth doing when a keyboard is also feeding a DAW and you do not want its notes firing Clips.

Velocity sensitivity

A video Clip with audio can take its level from how hard you hit the pad. The control is **Velocity Sensitivity**, in the Clip's audio options next to Volume.

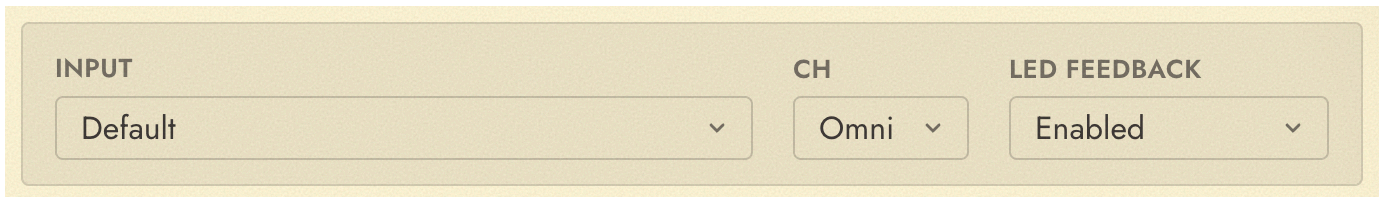
Volume stays the Clip's mix level. Velocity decides how far a hit moves it. At full sensitivity the response is the squared curve hardware samplers use; lower settings blend toward a fixed level, the way a synth's velocity-amount knob does. It is off on every Clip until you turn it on, so nothing about an existing show changes.

Triggers that carry no velocity of their own count as a full-strength hit: the mouse, the computer keyboard, a Stream Deck, the API, OSC, the command line and Song Timelines.

One limit worth knowing: if the Clip's Retrigger Action is Stutter, Reverse, Scratch, or Do Nothing, velocity applies to the first hit only. Visibox captions the slider "First hit only" when that is the case, so it does not read as broken.

MIDI Feedback

Controllers with lit pads can show you the state of the Song without you looking up. Visibox sends the mapped notes back out to the controller, so the pad you press to fire Clip 3 is the pad that lights up for Clip 3.



The screenshot shows a settings panel with three sections: INPUT, CH, and LED FEEDBACK. Each section has a dropdown menu. The INPUT dropdown is set to 'Default', the CH dropdown is set to 'Omni', and the LED FEEDBACK dropdown is set to 'Enabled'. Each dropdown has a small downward arrow icon.

Turn LED Feedback on for a device in Settings > MIDI.

To turn it on, open [Settings > MIDI](#), choose your device in the **Input** list, and set **LED Feedback** to **Enabled**. Visibox sends to the output port belonging to that same device, so there is nothing else to route.

The lights mean:

- **Solid**: the Clip is loaded in the current Song and ready.
- **Fast blink**: the Clip is playing.
- **Slow blink**: the Clip is paused.
- **Off**: that pad has no Clip in this Song, or no mapping at all.

Change Songs and Visibox runs a short flourish across the pads, lighting each available Clip in turn. It confirms that the new Song loaded, and shows which pads are live in it.

TIP

Feedback pays off most on grid controllers with per-pad lights, such as a Novation Launchpad or an Akai APC.

[Watch this video: https://youtu.be/hYWjDoU3IWY](https://youtu.be/hYWjDoU3IWY)

VIDEO TUTORIAL: Use MIDI to trigger videos, images, cameras, and audio with Visibox. [YouTube link](#)

Selecting Songs via MIDI Program Change

If your controller or your backing-track software sends program change messages, those can select Songs. Open [Settings > MIDI](#), pick the device that will be sending them, and scroll to the **Select Song via Program Change** row at the bottom of the table. The dropdown has three values.

- **Disabled:** program change messages are ignored.
- **Song Order:** the message selects by position. Program Change 0 selects the first Song, 1 the second, and so on. Reorder the Songs and the numbers follow the new order.
- **Program Identifier:** the message selects the Song holding that Program ID. Reorder the Songs and each one keeps its own number.

Program Identifier is the one to use for a set you rearrange. Assign a Song's ID from its editor panel: right-click the Song in the Controller and pick a number under **Program Identifier**. Songs already holding an ID are listed beside their number, anywhere in the Project: one that is not in the active Setlist reads "(not in setlist)". Choosing a number that is taken swaps the two Songs' IDs rather than creating a duplicate. New Songs get the next free ID on their own unless you turn off **Auto-Assign Program ID to New Songs** in [Settings > General](#).

NOTE

Program change messages run 0 to 127 on the wire, but plenty of hardware and software displays them as 1 to 128, which is how Visibox numbers Program Identifiers. If your device counts from 0 instead, turn on **Program Change numbers start at 0** in [Settings > MIDI](#) and the numbers match again. **Song Order** mode takes care of itself either way, because it selects by position rather than by a number you assigned.

Sending MIDI from a Song Timeline

MIDI goes out as well as in. A Song Timeline can hold MIDI cues that fire at a point in the Song and send to a real port on your machine: notes on, notes off, control change, program change, pitch bend, channel pressure, poly pressure, and SysEx. That is how you get Visibox to change a patch on a rack unit or arm a light board on cue, without a separate sequencer running alongside it. Turn on [Settings > General > Enable Beta Features](#) to see this.

MIDI Monitor

The MIDI Monitor is a diagnostic window showing messages as they arrive. It is the first place to look when a controller is not doing what you expect, because it shows whether Visibox is receiving anything at all.

All Devices ▾

All Types ▾

Pause

Clear

2 devices connected

16 messages

Time	Type	Ch	MIDI Data	Device
31:28.963	Program	1	Program: 5	IAC Driver Visibox
31:29.061	Note On	1	C3 (60) velocity: 100	IAC Driver Visibox
31:29.159	Note Off	1	C3 (60) velocity: 0	IAC Driver Visibox
31:29.258	Note On	1	D3 (62) velocity: 96	IAC Driver Visibox
31:29.353	Note Off	1	D3 (62) velocity: 0	IAC Driver Visibox
31:29.453	CC	1	CC7 value: 20	IAC Driver Visibox
31:29.553	CC	1	CC7 value: 36	IAC Driver Visibox
31:29.651	CC	1	CC7 value: 52	IAC Driver Visibox
31:29.749	CC	1	CC7 value: 68	IAC Driver Visibox
31:29.847	CC	1	CC7 value: 84	IAC Driver Visibox
31:29.945	CC	1	CC7 value: 100	IAC Driver Visibox
31:30.042	Note On	1	E3 (64) velocity: 110	IAC Driver Visibox
31:30.139	Note Off	1	E3 (64) velocity: 0	IAC Driver Visibox
31:30.235	CC	1	CC1 value: 64	IAC Driver Visibox
31:30.332	Note On	1	C1 (36) velocity: 127	IAC Driver Visibox
31:30.431	Note Off	1	C1 (36) velocity: 0	IAC Driver Visibox

The MIDI Monitor displays messages from connected devices as they arrive.

Open it from the menu bar, or with the keyboard:

- **Mac:** Visibox > MIDI Monitor..., or **⌘M**.
- **Windows:** File > MIDI Monitor..., or **Ctrl+Alt+M**.

The window lists:

Column	Description
Time	When the message arrived (minutes:seconds.milliseconds)
Type	Note On, Note Off, CC, Program and so on
Ch	The MIDI channel, 1 to 16
MIDI Data	The detail: note name, velocity, CC value
Device	Which device sent it

The dropdowns at the top filter by device and by message type. **Pause** freezes the list so you can read it, and **Clear** empties it.

 TIP

Work through it in order. If pressing a pad shows nothing, the device is not connected or is turned off in Settings > MIDI. If messages appear but nothing fires, the note or CC number does not match what is in the map on that same category.

Real-Time Control

Most of what Visibox does is on or off. You press a pad and a Clip plays. **Real-Time Control** is the other half: a knob you turn, a fader you ride, a value that moves while the Clip is on screen. Fade a logo up over eight bars. Ride a blur through the breakdown. Scrub a Clip's playhead by hand.

It works through eight named **Controls**, lettered **A** through **H**. A Control is a slot, not a thing in itself. You point it at something in your Project, then you drive it from a knob on your controller, from a phone running an OSC app, or from a fader in the app itself.

What a Control Is

Every Project has Controls A through H from the start. Each one carries:

- A **name**, so you can call it “Blur” instead of “C”.
- A **color**, which shows next to the parameters it drives.
- A list of **Targets** it is pointed at. One Control can drive several at once.

Controls A through H are permanent. You can add more if you run out. A new one takes the letter after the highest in the list, so Visibox does not hand out a letter freed up from the middle again. That matters because a MIDI map and an OSC path both address a Control by its letter, and a reused letter would quietly drive the wrong thing.

A Control does nothing until you point it at something. An unused Control costs you nothing and changes nothing.

Targets

A **Target** is one parameter a Control can move. Three kinds exist.

Clip parameters, on any Clip:

Target	What it moves
Opacity	How solid the Clip is, from invisible to full.
Speed	Playback rate, on video Clips.
Position	The playhead, on video Clips. Scrub by hand.
Filters	Blur, Brightness, Contrast, Grayscale, Rotate Hue, Invert, Saturate, and Sepia.

Visualizer inputs, on an ISF (Interactive Shader Format) visualizer Clip. See [Visualizers](#) for the two Visualizer engines. Sliders, on-off switches, whole-number inputs, drop-down choices, and trigger inputs can all be driven. Two kinds cannot. Color and point inputs hold more than one number, and a Control carries one, so pointing a Control at either would move a single component and pin the rest. Image and audio inputs are the shader's source material rather than a setting, so they are not offered.

Effect inputs, on any Effect layered onto a Clip. These are the same parameters you see in the Clip's Tweak panel.

Targets belong to Clips, which means they belong to Songs. A Control pointed at a Clip in Song 3 does nothing while Song 1 is up.

Pointing a Control at a Target

You do this from the Clip, not from the settings window. Right-click a Clip to open its editor panel, then open the sub-panel holding the parameter you want:

1. Right-click the Clip and choose **Edit Clip...**
2. Open **Speed**, **Position**, **Opacity**, **Filters**, or **Tweak**, depending on what you want to drive.
3. Next to the parameter is a small pop-up menu, showing a bare dash until something is assigned; its tooltip names it **Assign to Control**. Choose a Control from it.
4. A colored dot appears beside the parameter, matching the Control you picked.

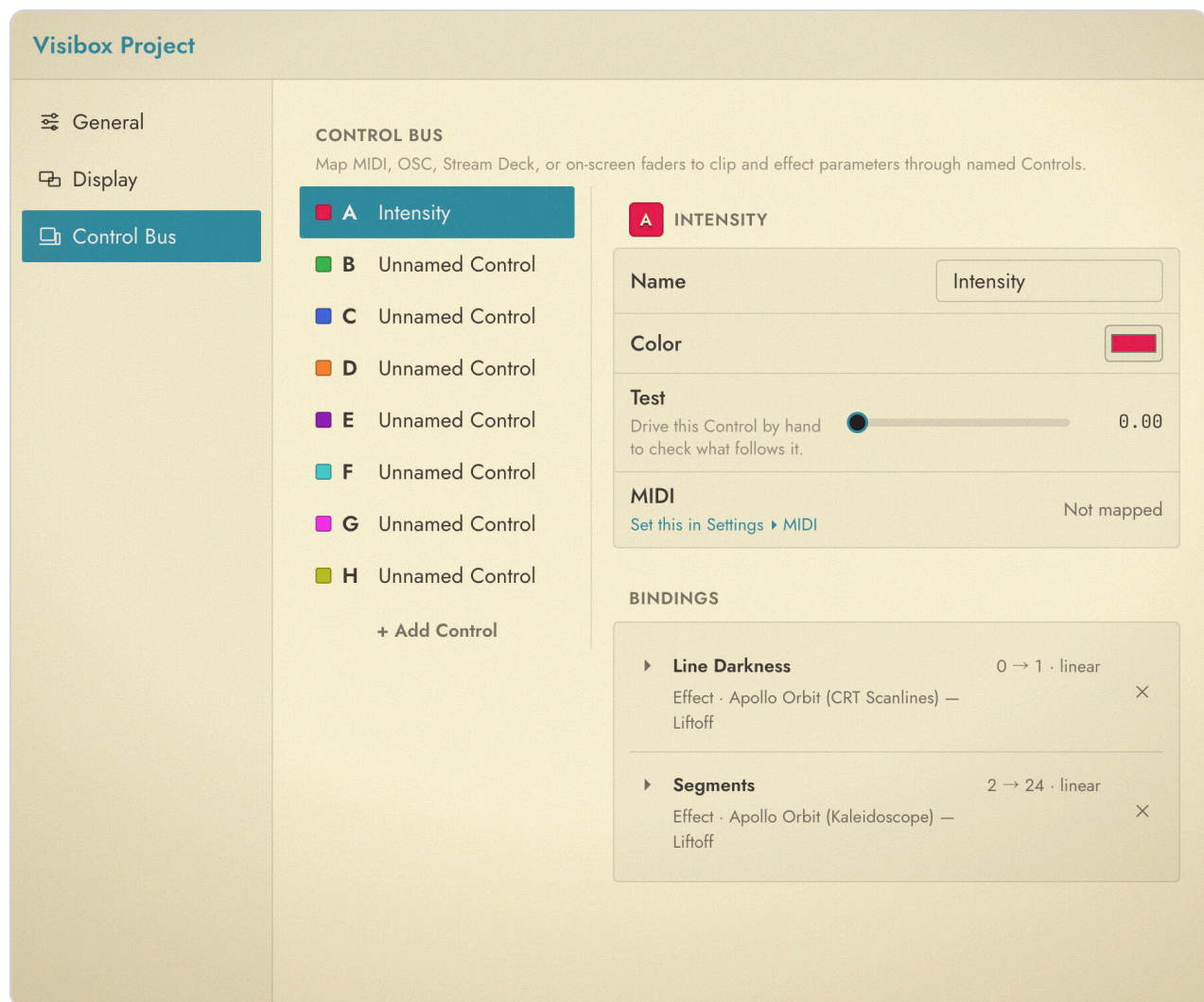
To unpoint it, choose the dash at the top of that same pop-up menu.

Select several Clips first and the assignment lands on all of them at once. If they disagree about which Control they are on, the menu shows **Mixed** until you pick one.

See [Clip Editor Panel](#) for the rest of what those sub-panels hold.

The Control Bus Tab

Once you have pointed a Control at something, the **Control Bus** tab in the Project Settings window is where you name it, test it, and shape how it responds. Choose **Project > Project Settings...** and pick **Control Bus** from the list on the left.



The Control Bus tab in Project Settings.

Down the left is every Control in the Project, each with its color and letter. Below the list, + **Add Control** adds another one. Pick a Control and its detail appears on the right:

Row	What it is
Name	What you call this Control. Leave it blank and it stays the bare letter.
Color	The color shown beside every parameter this Control drives.
Test	A fader you can drag to drive the Control by hand and watch what follows it.
MIDI	What is mapped to this Control, or Not mapped . Click through to set it up.
Bindings	Every Target this Control is pointed at, grouped by the Clip and Song it is in.

The Test fader is live, and it moves on its own when a hardware knob drives the same Control. It is the fastest way to check a patch before a show without touching the controller.

Removing a binding here unpoints that one Target and leaves the rest alone. Controls A through H cannot be removed, but any Control you added yourself can be.

Driving a Control

From a MIDI controller

A knob or fader that sends Control Change messages is the usual way in. In the MIDI Map there is a row per Control, reading **Control A**, **Control B**, and so on. Arm the row, move the knob, and it is mapped. See [Using MIDI](#) for the step-by-step.

A CC runs 0 to 127, and Visibox spreads that across the Control's full travel. Notes and buttons are not offered here: a Control wants something continuous.

The MIDI map belongs to your rig rather than to a Project, so the same knob drives Control A in every Project you open.

From OSC

Send a float to `/visibox/control/A`, replacing the letter as needed. Addressing by letter rather than by name means a phone layout you built last year still works against a Project you made this morning. See [OSC](#).

You can also write a Target directly, skipping Controls altogether, with `/visibox/param/`. That is on the OSC page too.

From a Stream Deck

The plugin's **Control** action sets or nudges a Control from a button, and sweeps it from a dial. See [Stream Deck](#).

From the command line and AI tools

`visibox control A 0.5` sets a Control from a shell or a script, and the API's MCP tools give an AI assistant the same reach. See [Command Line](#) and the [Visibox API](#).

From the app

The **Test** fader on the Control Bus tab. Useful for setup and for a show where you are already at the keyboard.

Shaping the Response

A raw knob is rarely what you want on stage. Each binding, not each Control, carries its own conditioning, so the same Control can move one Target across its whole range and nudge another gently.

Click a binding to open it. You get **min** and **max**, and a **curve**:

Curve	What it does
linear	Straight through. The knob and the parameter move together.
exp	Slow at the bottom, fast at the top. Good for opacity and volume.
log	Fast at the bottom, slow at the top. Good for a blur you want to feel early.

Curve	What it does
S	Gentle at both ends, quick through the middle.

Set **min** higher than **max** and the Control runs backwards, which saves rewiring a fader you have mounted upside down.

Behind the **Conditioning options** button are four more:

Option	What it does
deadzone	Ignores the bottom of the knob's travel and spreads the rest across the full range. Kills the drift on a cheap fader that never quite reaches zero.
smoothing	Eases the value over a number of milliseconds instead of jumping. Takes the stepping out of a coarse controller.
steps	Quantizes the value into a fixed number of stops. Two steps makes a switch out of a fader.
takeover	Decides what happens the moment a knob starts sending again. See below.

Takeover

A knob has a physical position. The parameter has a current value. When you switch Songs or reload a Project, those two disagree, and the first touch of the knob can throw the parameter somewhere you did not want.

Takeover	What happens on the first message
None	The Target jumps straight to wherever the knob is.
Pickup	Nothing happens until the knob passes through the Target's current value, then it takes over.
Scale	It takes effect at once, with the knob's remaining travel rescaled so its current position maps to the Target's current value.

Pickup is the safe one for a show, and the reason most desks offer it. **Scale** is the one for a soft controller with a motorless fader you want responding immediately. **None** is fine while you are building a Project.

Tips



TIP

Name your Controls before the second gig, not after. "Blur" and "Logo Fade" are readable in a dark room. "C" and "F" are not.

NOTE

A Control set up in one Project does not carry into another, because it is pointed at that Project's Clips. What does carry is the MIDI mapping: the same knob lands on Control A wherever you open it, ready for whatever that Project has pointed A at.

See Also

- [Using MIDI](#) for mapping a knob to a Control.
- [OSC](#) for driving Controls from a phone, a tablet, or a cue system.
- [Clip Editor Panel](#) for the sub-panels where bindings are made.
- [Effects](#) for the Effect inputs a Control can drive.

Connecting an Elgato Stream Deck

Visibox drives an Elgato Stream Deck through a plugin that runs inside Elgato's Stream Deck software. Drop a Visibox action onto a button and it triggers a Clip, switches Songs, steps through the set, or sets the output volume. Clip buttons carry the Clip's own thumbnail, so the deck reads like the Song in front of you.



A Stream Deck running the Visibox plugin with live clip thumbnails.

Because the plugin lives in the Stream Deck software, Visibox actions and the actions from your other apps share the same device, and you lay out your buttons with the tools you already know.

Coming from Visibox 5

Visibox 6 no longer drives a Stream Deck over USB by itself. The old built-in mode and the **Stream Deck Mode** setting are gone, and the plugin is the only path.

If your deck went quiet right after you updated, this is why. Install the plugin as described below and your buttons come back.

A few things follow from the change:

- **Elgato's Stream Deck software needs to be running.** Under the old built-in mode you had to quit it. Now it is required.
- **Buttons you built in the Stream Deck software still work.** If you were already using the plugin in 5.x, nothing changes.
- **The old built-in layout is not converted.** Visibox used to take over every button and fill them with the current Song's Clips plus Stop, Previous Clip, Next Clip, Previous Song and Next Song. You rebuild that layout from the actions below, or start from one of the bundled profiles.

Here is the built-in layout translated into plugin actions:

Old built-in button	Plugin action
A Clip in the current Song	Trigger Clip , with Clip Number set to its position
Stop	Playback Control , Control Type Stop
Previous Clip / Next Clip	Navigation , Previous Clip / Next Clip
Previous Song / Next Song	Navigation , Previous Song / Next Song

Before you start

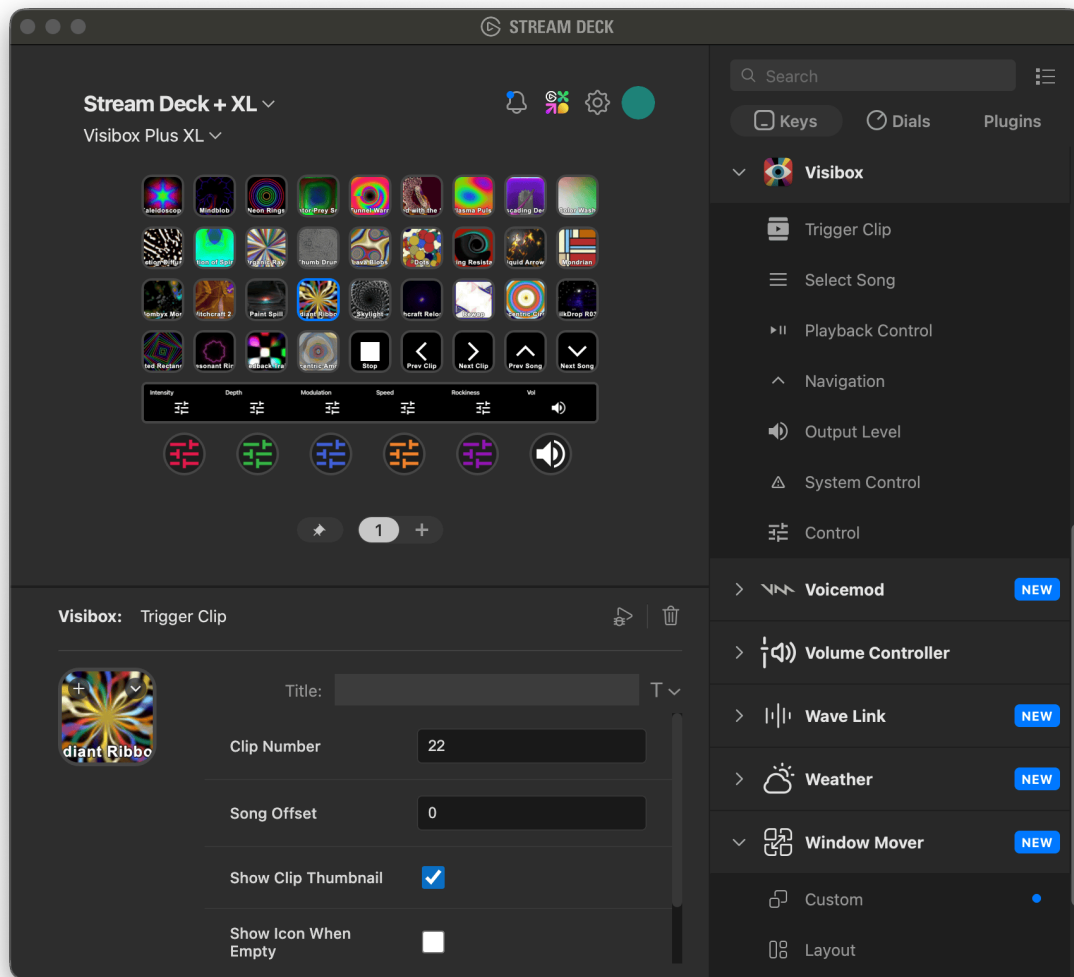
- Elgato's **Stream Deck software, version 6.9 or later**, installed and running.
- Visibox running on the same computer. The plugin talks to Visibox on that machine, so Visibox has to be open for the buttons to do anything.
- The Stream Deck software itself needs macOS 10.15 or later, or Windows 10 or later.

Installing the plugin

Visibox ships with a copy of the plugin, so you install it from inside Visibox.

1. Open **Settings > Devices** in Visibox.
2. In the **Stream Deck** section, find **Stream Deck Plugin** and click **Install Stream Deck Plugin....**
Visibox hands the bundled plugin file to Elgato's Stream Deck software.
3. Confirm the install when the Stream Deck software asks.

Open the Stream Deck software and you will find a **Visibox** category in the action list on the right.



Visibox actions in the Stream Deck software. Drag an action onto a button to configure it.

Available actions

The plugin provides seven action types that you can drag onto any Stream Deck button.

Action	Description
 Trigger Clip	Trigger a Clip by its position in the current Song. The button displays a live thumbnail.
 Select Song	Switch to a Song by number, Program Identifier, or ID.
 Playback Control	Play, Pause, Stop, Resume, Play/Stop Toggle, or Pause/Resume Toggle.
 Navigation	Next/Previous Clip or Next/Previous Song.
 Output Level	Set, nudge, or dial the output volume, and toggle mute.
 System Control	Panic (Reset) or Fullscreen Toggle.

Action	Description
 Control	Set or nudge one of the Project's Real-Time Controls by its letter.

Trigger Clip

Set **Clip Number** to the Clip's position in the Song, where 1 is the first Clip. The button displays the Clip's thumbnail and title, and pressing it triggers the Clip. This action always targets the active Project.

Song Offset changes which Song the **Clip Number** refers to, relative to the active Song. An offset of 0 (the default) targets a Clip in the active Song, -1 the previous Song, and 1 the next Song. Use it to pre-cue the next Song while the current one is playing.

The button also respects the Clip's [Launch Mode](#). A Clip set to **Gate** plays while you hold the button down and stops when you let go. **Trigger** and **Toggle** Clips ignore the release, so they behave the same as they do from any other trigger source.

By default the button shows the Clip's thumbnail. Uncheck **Show Clip Thumbnail** to display the plain action icon instead, or check **Show Icon When Empty** to show an icon on Clip slots that have nothing assigned.

Select Song

Fill in exactly one of three fields:

- **Song Number** selects a Song by its position in the Project, where 1 is the first Song.
- **Program Identifier** selects the Song with that MIDI Program Identifier, 1 to 128.
- **Song ID** selects a Song by its internal identifier. It survives reordering, which makes it the reliable choice for a fixed show.

If you fill in more than one, Song ID wins, then Program Identifier, then Song Number. The button displays the Song's title.

Stop Current Playback is on by default and stops any playing Clips before switching. Turn it off to let Clips keep playing across the Song change.

Playback Control

Choose a Control Type: Play, Pause, Stop, Resume, Play/Stop Toggle, or Pause/Resume Toggle. The button icon follows the current playback state.

Navigation

Pick a direction: **Next Clip**, **Previous Clip**, **Next Song**, or **Previous Song**.

For Clip navigation, **Unconstrained** lets navigation cross Song boundaries, wrapping from the last Clip of one Song to the first Clip of the next.

For Song navigation, **Don't Stop Playing Clips** keeps the current Clips playing as you switch Songs.

Output Level

This action controls Visibox's output volume. **Control Type** decides what a press does:

- **Set to Level** jumps straight to the **Volume Level** you enter, from (silent) to (full). Leave **Volume Level** blank and the press toggles mute instead.
- **Nudge Up** and **Nudge Down** move the live volume by one step, the same way the **Volume Up** and **Volume Down** menu items do. The level snaps to round values, so repeated presses land where you expect, and it stops at 0.0 and 1.0.

Step Size sets the amount per press for the two nudge types. It defaults to 0.1, or 10%, matching the Volume Up and Volume Down menu items.

On a Stream Deck with dials, drop this action onto a dial. Rotating the dial adjusts the volume and pressing it toggles mute, whatever Control Type you picked. The dial moves by 0.05 per tick by default, and **Step Size** changes that too, up to a ceiling of 0.2 per tick.

System Control

Panic (Reset) reloads all of Visibox's windows. It is the button to reach for when something goes wrong mid-show.

Fullscreen Toggle switches the output window in and out of full screen. The button reflects the current state, and it follows along when you change full screen from the menu or the keyboard as well.

Control

Drives one of the Project's [Real-Time Controls](#) by its display letter, A to Z. Set the Control to an exact value, or nudge it up and down by a step you choose; values run 0 to 1. The button shows the name you gave the Control in the Control Bus, or its letter until you name it. Addressing a letter no Control uses does nothing.

On a Stream Deck with dials, the action works on a dial too: rotate to sweep the Control, press or touch to set or nudge it.

Bundled profiles

The plugin ships with a ready-made profile for each Stream Deck model, sized to that model's grid:

Profile	Device
Visibox Mini	Stream Deck Mini
Visibox Performance	Stream Deck and Stream Deck MK.2
Visibox XL	Stream Deck XL
Visibox Plus	Stream Deck +
Visibox Neo	Stream Deck Neo
Visibox Studio	Stream Deck Studio

Profile	Device
Visibox Plus XL	Stream Deck + XL

Most profiles fill the grid with **Trigger Clip** buttons and give the last keys to transport. Visibox Performance, Visibox XL, Visibox Studio and Visibox Plus XL end with five transport keys: Stop, Previous Clip, Next Clip, Previous Song and Next Song. Visibox Plus and Visibox Neo have room for three, Stop, Previous Clip and Next Clip. Visibox Mini is all **Trigger Clip** keys, with no transport.

On the models with dials, Visibox Plus, Visibox Studio and Visibox Plus XL, the first dial is wired to **Output Level**. The rest are left empty for you to fill.

Profiles arrive with the plugin and wait in the Stream Deck software's profile list. They do not take over the profile you are using, so you switch to one when you want it. A profile is a good starting point if you are rebuilding a layout from Visibox 5, or if you want a working set of buttons before designing your own.

Live thumbnails

A **Trigger Clip** button shows a live thumbnail of the Clip on the button face. Thumbnails follow the active Song, so the buttons always show what they will play. Visualizer Clips render a real preview rather than a generic icon, and Color Clips show their assigned color.

Camera Clips also show a thumbnail, refreshed about every five seconds rather than at full frame rate.

Multiple Stream Decks

You can use more than one Stream Deck with Visibox at the same time. Each device is configured independently in the Stream Deck software, so you can give each one its own profile. A Stream Deck XL for Clip triggering and a Stream Deck Mini for Song navigation and transport is a common split.

Languages

The plugin follows the language you set in the Stream Deck software: English, German, French, Spanish, Japanese, and Korean. Action names and the settings inside each action's panel are both translated.

Troubleshooting

No Visibox category in the action list:

- Check that the Stream Deck software is version 6.9 or later.
- Reinstall the plugin from **Settings > Devices > Install Stream Deck Plugin...**, then quit and reopen the Stream Deck software.

Install Stream Deck Plugin... reports an error:

- Install Elgato's Stream Deck software first. Visibox hands the plugin file to that software, so there is nothing to open it without it.

Buttons appear but nothing happens when you press them:

- Start Visibox. The plugin connects when Visibox launches, and keeps trying every five seconds for as long as Visibox is open.

A thumbnail stays blank:

- Remove the action from the button and drag it on again.

Stream Deck disconnects during a show:

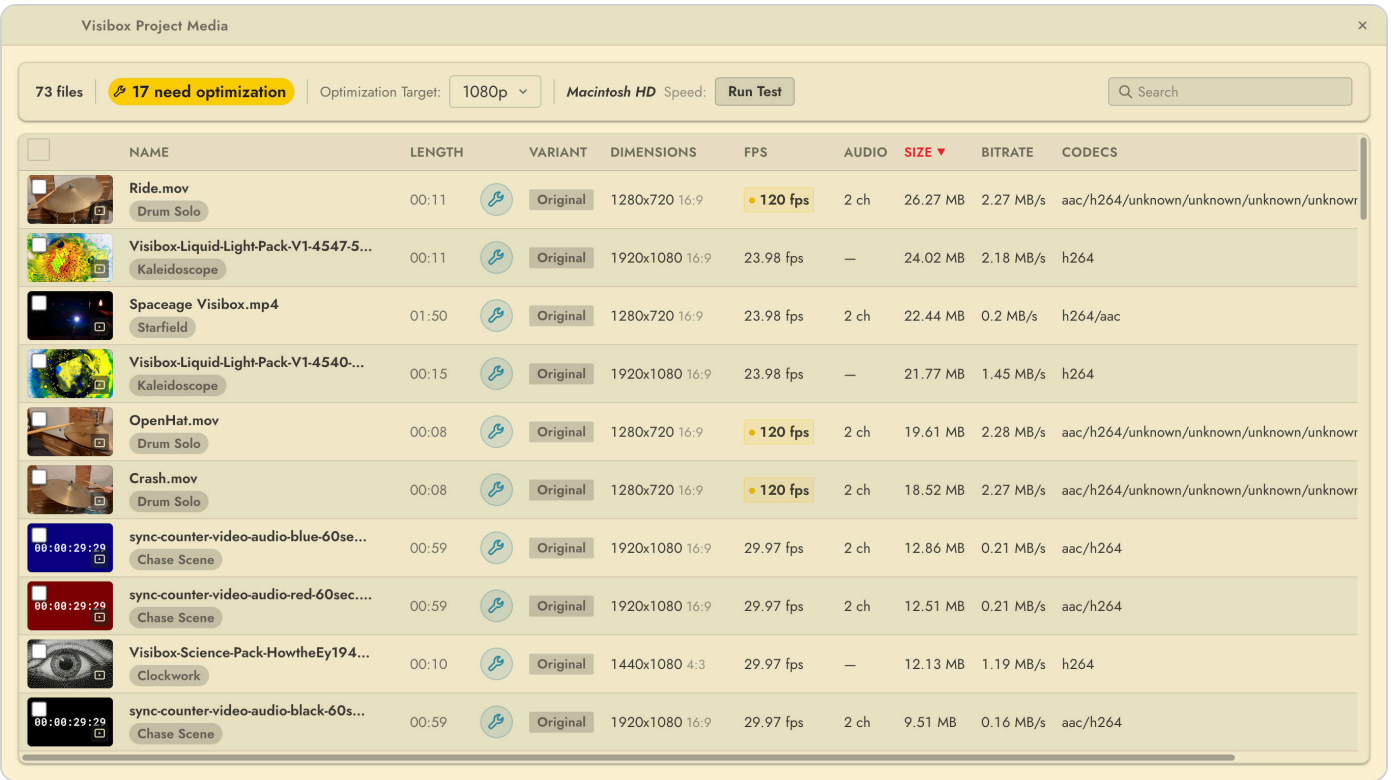
- After a brief disconnect, a bumped USB cable for instance, the plugin reconnects on its own. It retries every five seconds while Visibox is running, and the buttons come back with their current states.
- For reliable connections during a performance, use a short, good-quality USB cable and avoid hubs where you can.

Media Management & Optimization

Visibox plays most media as it comes, but files often arrive heavier than the show needs: a poster-sized image, a video with a cinema bitrate, a 96 kHz audio file playing through a 48 kHz system. Optimizing re-encodes those files so they load faster and play back smoothly, and it repairs bad encodes that would otherwise pause or glitch. The Media Overlay is where you see which files are worth the trouble, and fix them in place.

The Media Overlay

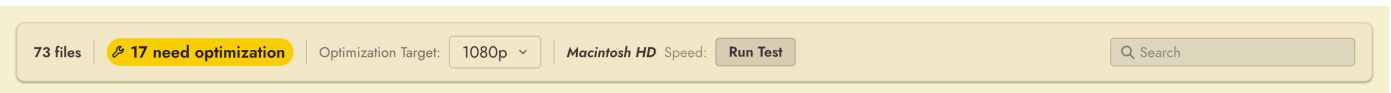
Choose **Project > Manage Media...** to open the Media Overlay. It lists every media file in the current Project.



The Media Overlay lists, renames and optimizes your media files. It is also where a video or image gets a custom thumbnail.

To close it, click the close button in the upper right corner or press Escape.

The Information Bar



The Information Bar

The bar across the top sums up the Project's media, and most of its values do something when clicked. Hover over a value for the detail.

Item	What it shows and does
File count	How many media files the Project holds.
Selected	How many files are selected, shown only while something is. Its tool button optimizes the selected files.
x need optimization	How many files are out of range. Its tool button optimizes them all; clicking the text itself filters the list down to them.
Optimization Target	The size and format files are converted to when you optimize. Click it to choose 720p, 1080p, 4K, or 8K. The default is 1080p. A file that exceeds the target's dimensions, frame rate, or bitrate counts as needing optimization.
Drive speed	How fast the drive holding the Project reads, which is what decides whether big files load in time. Click Run Test to measure it, and click the result to run the test again.
Search	Filters the list. It matches any part of a name, a type, and most other metadata, ignoring case. Click the "x" to clear it.

If the test reports a slow drive, move the Project to a faster one or optimize the media so it asks less of the drive it is on.

The columns

☐	NAME	LENGTH	VARIANT	DIMENSIONS	FPS	AUDIO	SIZE ▼	BITRATE	CODECS
--------------------------	------	--------	---------	------------	-----	-------	--------	---------	--------

The Media Table Header

Click any column header to sort by it.

Column	What it holds
Media Type	The thumbnail or waveform, icons naming the type, the selection checkbox, and (for images and videos) the custom thumbnail button. The checkbox in the header selects or deselects every visible file.
Name/Songs	The file's name and the Songs that use it. Click the name to rename the file.
Duration	The length of a video or audio file. An image reads ∞.
Optimization	The file's optimization status. Sorting here groups optimized and unoptimized files together.
Variant	The variant files that exist for this media. An optimized file shows both the original and the optimized copy.
Dimensions	Width by height in pixels, and the aspect ratio.
FPS	A video's frame rate.
Size	The file size in megabytes. An optimized file shows the original size too.
Bitrate	The file's bitrate in MB/s.

Column	What it holds
Codecs	The video and audio codecs in the file. For an image, the file format.

Working in the list

	Ride.mov Drum Solo	00:11		Original	1280x720 16:9	120 fps	2 ch	26.27 MB	2.27 MB/s	aac/h264/unknown/unknown/unknown/unknown
	Visibox-Liquid-Light-Pack-V1-4547-5...	00:11		Original	1920x1080 16:9	23.98 fps	—	24.02 MB	2.18 MB/s	h264
	Spaceage Visibox.mp4 Starfield	01:50		Original	1280x720 16:9	23.98 fps	2 ch	22.44 MB	0.2 MB/s	h264/aac
	Visibox-Liquid-Light-Pack-V1-4540-...	00:15		Original	1920x1080 16:9	23.98 fps	—	21.77 MB	1.45 MB/s	h264
	OpenHat.mov Drum Solo	00:08		Original	1280x720 16:9	120 fps	2 ch	19.61 MB	2.28 MB/s	aac/h264/unknown/unknown/unknown/unknown
	Crash.mov Drum Solo	00:08		Original	1280x720 16:9	120 fps	2 ch	18.52 MB	2.27 MB/s	aac/h264/unknown/unknown/unknown/unknown

The Media Table Body

Each row is one media file. Hover over anything in a row for a tooltip.

Selecting. Click a file's checkbox or its thumbnail to select it, and shift-click to select a range. Selected files show a blue outline, and the Information Bar offers to optimize them as a group.

Renaming. Double-click a file's name, or click its edit button, then press Enter to save or Escape to cancel.

Optimizing. A file that needs optimization has a live tool button in its Optimization column; click it to convert the file to the Optimization Target. Hold Shift while clicking to optimize every unoptimized file at once. Values that exceed the target are highlighted in yellow so the files worth attention stand out in a long list, and clicking a highlighted value optimizes that file directly.

After optimizing. The row splits to show the original and the optimized copy side by side, sizes and all, with the unused original in a lighter color. The tool button becomes a revert button that puts the original back. Nothing is thrown away.

Custom thumbnails. A video or image can carry a thumbnail of your choosing: click the button in the Media Type column and pick an image, or drag an image straight onto the existing thumbnail. The same button then removes it. Custom thumbnails earn their keep in a Project full of similar-looking files, and in a Stream Deck kiosk where visitors pick media by picture.

HEVC proxies

HEVC (H.265) video plays as it comes on almost every machine. Where Windows cannot decode it in hardware, Visibox plays an H.264 copy instead. The copy is a proxy: your original stays in the Project untouched, and the proxy sits in the list above like any other optimized copy, revert button included.

Opening a Project that holds HEVC video on such a machine brings up **Optimize HEVC videos for playback?**, offering **720p**, **1080p** and **4K**. The size you pick becomes the Project's Optimization Target, and every HEVC file in the Project is queued. **Cancel** queues nothing, and those Clips will not play until you optimize them by hand from the list above. Files you add afterwards are converted without asking.

Conversion runs in the background and waits while anything is playing on the Output, so an encode never competes with a show. A Clip whose proxy is not ready does nothing when triggered, and plays as soon as the proxy lands.

Keeping the indexer out of your media

A Project folder is a hard place for a file indexer to sit. Every save rewrites the Project file, every import copies gigabytes of video in, thumbnails are written continuously, and a fresh backup lands every few minutes. Each of those sends the indexer back over the folder, and that disk work competes with playback on the machine running your show.

On macOS, Visibox marks a Project's media folder and its Backups folder to be skipped by Spotlight. This happens on its own when you open a Project. There is nothing to switch on.

The Project file itself is deliberately left indexed, so your Project still turns up in a Spotlight search by name. Only the bulk media and backups inside it are skipped.

If the folders sit on a read-only volume, or permissions get in the way, Visibox carries on without the exclusion rather than failing to open the Project.

Windows has no equivalent per-folder opt-out, so Visibox offers instead to add a Windows Defender exclusion for the Project folder. See [Support & Bug Reports](#).

Effects

Effects are real-time visual treatments applied to a Clip. They add motion, texture, distortion, color shifts, and audio-reactive behavior without touching your original media files. A Clip can carry several at once, and every Effect can be dialed in for that one Clip.

Two kinds of Effect

Visibox runs two kinds of Effect side by side, and you use them the same way.

Kind	What it is
CSS effect	Built from CSS and optional SVG filters. Good for motion, color grading, blurs, retro looks, and anything that treats the whole frame.
ISF effect	A GLSL fragment shader in Interactive Shader Format . Good for warps, feedback, and per-pixel math.

The difference that matters in practice is where each one sits. An ISF effect runs on the Clip's source media. CSS effects wrap what comes out of that. So an ISF warp followed by a CSS color grade grades the warped image, not the original.

Applying Effects to a Clip

Right-click a Clip and open the **Effects** row, or choose **Clip > Edit Clip....** The row names everything currently on the Clip, and its panel is the Clip's effect stack.

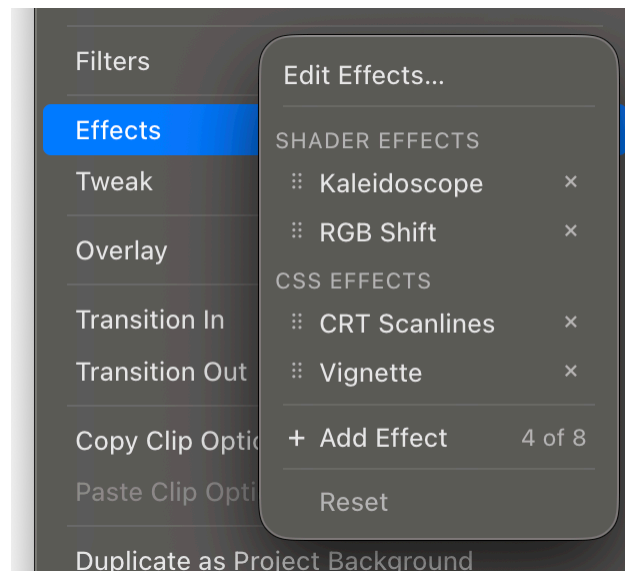


Right-click a Clip and open the Effects row.

Click **Add Effect** and a picker opens. Type in **Search effects...** to narrow the list, then click an Effect and it applies immediately. If the Clip is playing you see it in the Output right away. An Effect already on the Clip is left out of the picker, since each one goes on a Clip once.

The effect stack

The panel lists the Effects on the Clip in the order they run, top to bottom. A counter beside **Add Effect** reads **3 of 8**, or whatever the count is: a Clip carries up to eight Effects. At eight the panel reads **Effect limit reached** and **Add Effect** does nothing until you take one off.



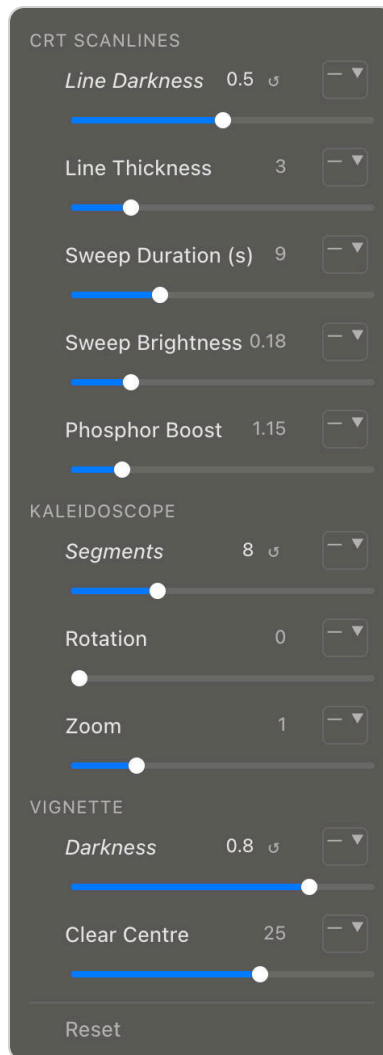
Several Effects on one Clip, in the order they run.

- **Drag to reorder.** Once two or more Effects of the same kind are on the Clip, a grip appears at the left of each row. Drag a row up or down to change where it sits in the stack.
- **Remove** one with the × at the right of its row. **Reset** at the bottom clears the lot.
- **Edit Effects...** at the top opens the Effects editor.

When a Clip holds both kinds, the panel groups them under **Shader effects** and **CSS effects**. The groups are the render order: an ISF effect runs on the source media and CSS effects wrap what comes out, so a shader is always inside a CSS effect. You can reorder within a group, but not drag a row across the boundary.

Tweak

Under the Effects row is a **Tweak** row. It holds the parameters that the Clip's Effects expose: a speed, an amount, a color, whatever the preset's author declared.



Tweak holds the parameters this Clip has changed.

Tweak changes apply to this Clip only. Two Clips using the same Effect can hold completely different values, and editing the preset later still reaches every parameter neither of them overrode. The row shows a count of how many parameters this Clip has changed.

The Tweak row appears only when the Clip's Effects actually declare parameters. An Effect with nothing to adjust shows no row rather than an empty panel.

To change the values everything starts from, edit the Effect's **Preset Defaults** in the Effects editor instead. Those apply to every Clip that has not set its own.

Audio Source

Audio-reactive Effects respond to an audio signal, and you choose which one per Clip.

The Audio Source row is always there for a Visualizer Clip, since a Visualizer is audio-reactive from top to bottom. An Audio Clip gets it once a Visualizer is paired with it. On a video, image, or camera Clip the row appears once the Clip has an Effect on it. An Effect is the only other thing on those Clips that can read the audio.

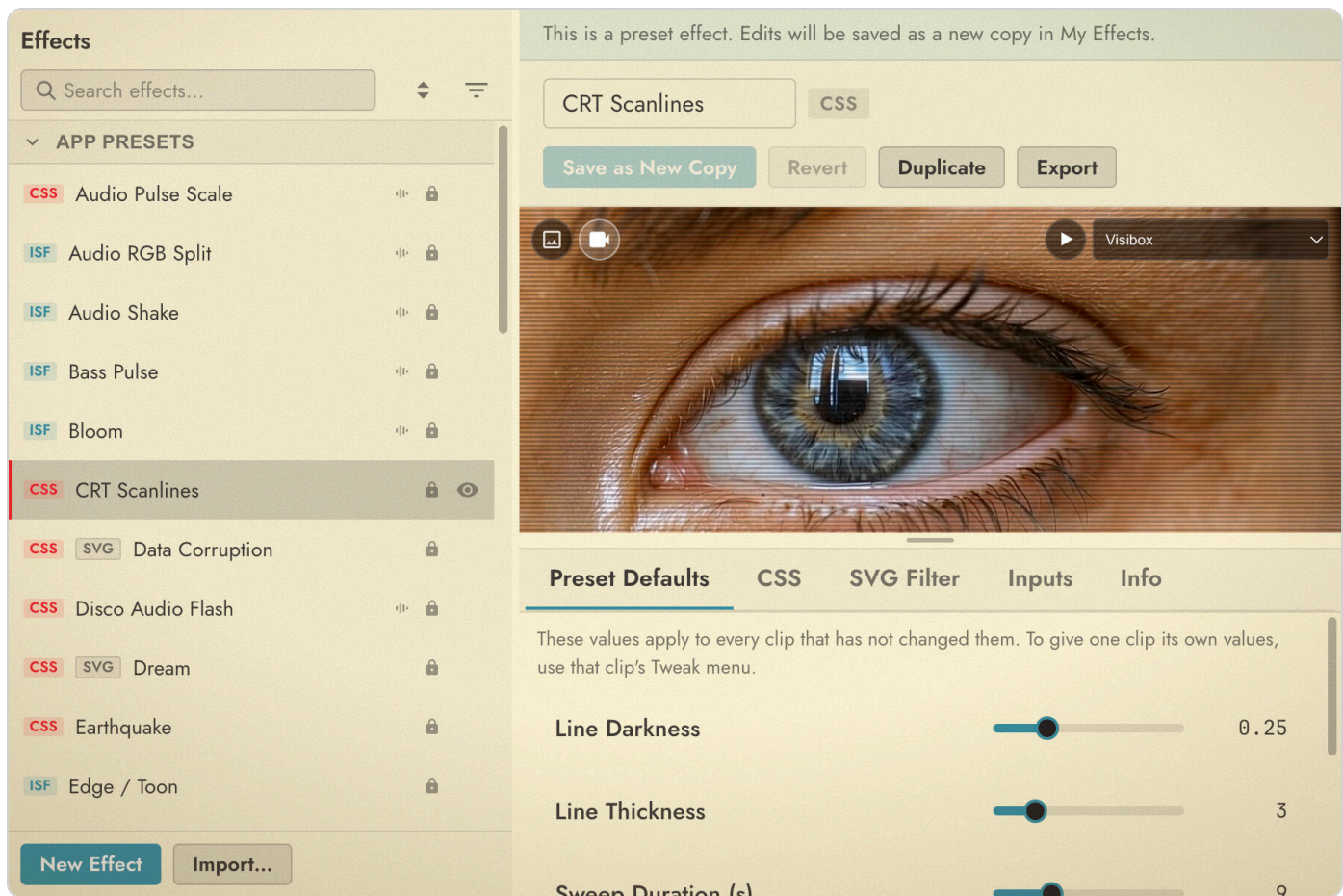
Right-click the Clip and open the **Audio Source** row:

Option	What it listens to
None	Nothing. The Effect still runs but stops moving with the music.
This Clip	This Clip's own sound and nothing else. Audio Clips only.
Visibox	The audio playing inside Visibox: Clip audio, Song audio, and the master mix. This is the default.
System Audio	Audio from other applications on the computer, such as a streaming service, a DAW, or a browser.
an input device	A microphone, a line input, or an audio interface. Your available inputs are listed by name below the options above.

System Audio is listed only once System Audio Input is switched on in Settings > Audio. On macOS 14.2 and later, the first time you enable it macOS asks for permission to capture system audio.

The Effects editor

The Effects editor is a tab in the [Editor Window](#). Open it from **Effects...** in the Visibox menu on Mac or the File menu on Windows.



The Effects editor: browse presets on the left, edit and preview on the right.

Where Effects come from

The list is grouped into four sections:

Section	What is in it
App Presets	The Effects that ship with Visibox. Read-only.
Plugins	Effects loaded from your Plugins folder at startup. Read-only.
My Effects	Effects you created or imported. Fully editable, stored on your machine.
Project Effects	The copies stored inside the Project you have open. These are what actually run.

Linked and unlinked

Applying an App Preset, a Plugin, or one of My Effects to a Clip puts a **linked** copy in the Project. A linked copy:

- **Stays in sync.** Edit the original in My Effects. Every Project holding a linked copy of it picks up the change the next time you open it, as long as the original is still on the same computer.
- **Cannot be edited in place.** Edits are saved as a new copy instead.

Editing a linked Effect never touches the original: the editor says so, the Save button reads **Save as New Copy**, and saving puts an independent, editable copy in My Effects. Right-clicking the Effect under Project Effects and choosing **Duplicate to My Effects** does the same thing without opening the editor. Either way, later changes to the original leave the copy alone.

Projects are self-contained

Every Project stores a complete copy of its Effects, linked ones included:

- **Deleting one of My Effects will not break a Project.** The Project falls back to its own copy and plays exactly as before. Visibox notes that the original source is gone, but the visuals do not change.
- **Projects travel with their Effects.** Your `.vsbx` Project carries everything it needs to play back on another machine.

Browsing

- **Search** by name in the box at the top.
- **Kind badges** on every Effect in the list mark it as CSS or ISF.
- **Filter** to CSS or ISF effects, by section, or by visibility.
- **Sort** alphabetically or by section.
- **Hide** Effects you never use with the visibility toggle that appears on hover.

Changes to the list can be taken back while the editor tab is showing. Press `⌘Z` or use **Edit > Undo**, and the Edit menu names the step: **Undo Hide Effect**, **Undo Show Effect**, **Undo Remove Effect** and so on.

Apply to Selected Clips

Select one or more Clips in the Controller, then right-click an Effect in the editor. The first row of the menu applies it to everything you selected, and names the count: **Apply to 3 Selected Clips**. Faster than working through them one at a time.

Audio audition

To hear how an Effect behaves, hover its preview and click the play button that appears. Visibox plays a test signal so you can watch the audio-reactive behavior while you design, with no Project needed.

The editor tabs

Tab	What it is
Preset Defaults	The values every Clip starts from. A Clip's own Tweak values override these.
CSS	The stylesheet, for a CSS effect. Use <code>.effect-target</code> as the selector and <code>effect-anim</code> as the keyframe name.
SVG filter	An optional SVG filter for a CSS effect: blur, displacement, channel manipulation.

Tab	What it is
ISF Shader	The GLSL fragment shader, for an ISF effect. The Clip's source media arrives as the <code>inputImage</code> input.
Vertex	An optional vertex shader for an ISF effect. Leave it blank for the default pass-through.
Inputs	The parameters this Effect exposes. Declare them here and they show up in Preset Defaults and in Tweak.
Passes	Multi-pass rendering, for an ISF effect. Each pass can write to a buffer that a later pass reads.
Info	Name, description, credits, and categories.

CSS effects show the CSS and SVG filter tabs. ISF effects show ISF Shader, Vertex, and Passes.

Creating an Effect

1. Click **New** in the Effects editor to add an Effect to My Effects.
2. Double-click the title to name it.
3. Write the effect in its code tab.
4. Add anything you want adjustable to the **Inputs** tab, then set sensible starting values in **Preset Defaults**.

Editing an App Preset or a Plugin is allowed, but saving writes a new copy into My Effects rather than overwriting the original. The editor tells you so before you save.

Audio-reactive CSS properties

Visibox updates a set of CSS custom properties in real time from the Clip's chosen [Audio Source](#). Use them anywhere in your stylesheet.

Frequency bands (0.0 to 1.0, smoothed):

Property	Range
<code>--audio-bass</code>	20–250 Hz
<code>--audio-low-mid</code>	250 Hz – 1 kHz
<code>--audio-mid</code>	1–4 kHz
<code>--audio-high-mid</code>	4–8 kHz
<code>--audio-treble</code>	8–20 kHz

Peak values (0.0 to 1.0, fast attack, slow decay):

Property	Range
<code>--audio-bass-peak</code>	20–250 Hz peak

Property	Range
--audio-low-mid-peak	250 Hz – 1 kHz peak
--audio-mid-peak	1–4 kHz peak
--audio-high-mid-peak	4–8 kHz peak
--audio-treble-peak	8–20 kHz peak

Level meters (0.0 to 1.0):

Property	Description
--audio-level	Overall RMS level (mono)
--audio-level-l	Left channel RMS level
--audio-level-r	Right channel RMS level
--audio-peak	Overall peak level

Use them like any custom property:

```
.effect-target {
  transform: scale(calc(1 + var(--audio-bass-peak) * 0.15));
  filter: brightness(calc(1 + var(--audio-peak) * 0.5));
}
```

ISF shaders read audio through the format's own `audio` and `audioFFT` input types: the raw waveform and the frequency spectrum. See the [ISF documentation](#).

TIP

Audio-reactive Effects read best on music with clear dynamics: a strong beat, a defined bass line, or sharp transients. They also stand out more on darker footage, where the glow and pulse have somewhere to go.

Importing and exporting

Effects are shared as `.vbox` files. To import one, drop it into the Effects editor and it lands in My Effects.

To send one the other way, right-click it and choose **Export to Plugins Folder**. To install a file someone sent you, put it in your Plugins folder:

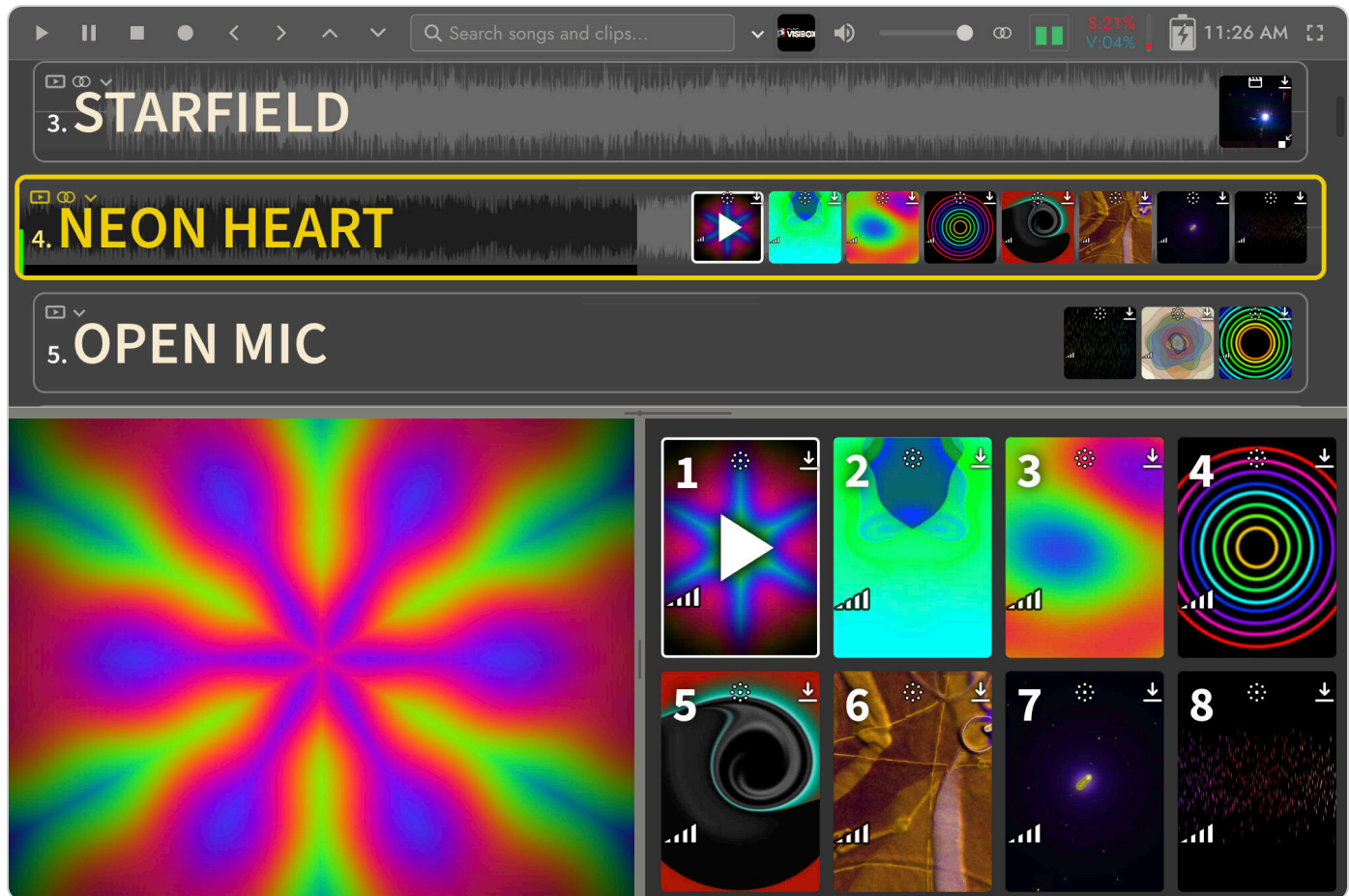
- **Mac:** `~/Library/Visibox/Plugins/`
- **Windows:** `%APPDATA%\Visibox Plugins\Plugins\`

Plugins load at startup and appear in the Plugins section of the list.

Visibox 6.0 moved the Windows folder. It used to be `%LOCALAPPDATA%\Visibox\Plugins\`, which sat inside the install and was wiped when Visibox was uninstalled, taking your plugins with it. Visibox moves what is there the first time 6.0 starts, so you have nothing to do.

Visualizers

Visualizers are live, generative, audio-reactive Clips: a built-in VJ engine that needs no media files. Visibox runs two engines, **ISF** and **MilkDrop**, each with its own history and its own look.



A Song of audio-reactive Visualizer Clips playing in the Controller window.

Adding a Visualizer Clip

1. Right-click a Song and choose **Add Visualizer Clip**.
2. A dialog lists the available presets. Browse or search for one you like.
3. Select it and click **Add**. The Clip appears in the Song alongside your others.

You can also drag a preset straight out of the Visualizers editor onto a Song in the Controller, or right-click a preset there and choose **Add to Active Song**.

Visualizer Clips carry a distinctive icon (⋮) on their thumbnail so you can tell them from video and image Clips.



A Visualizer Clip (note the icon) alongside video Clips in a Song.

Visualizer Clip options

Right-click a Visualizer Clip for its options.

Audio Source

Every Visualizer Clip picks the audio it reacts to, and so does an [Audio Clip](#) once you pair a Visualizer with it. Any other Clip gets the same row once it carries an Effect.

The options are the same wherever the row appears: **None**, **This Clip** (Audio Clips only), **Visibox** (the default), **System Audio**, or a named input device. See [Audio Source](#) in Effects for what each one listens to and when System Audio appears.

Tweak

Below the preset picker is a **Tweak** row holding the parameters this preset exposes. Values you set there belong to this Clip alone, so two Clips built from the same preset can look completely different. The row shows how many parameters this Clip has changed.

ISF presets declare their own inputs, so most of them have something to tweak. A MilkDrop preset declares no inputs at all, so a Clip running one shows no Tweak row.

ISF (Interactive Shader Format)

ISF was created by [Vidvox](#) in 2013 for their VDMX VJ software. The goal was a portable, reusable format for GLSL fragment shaders, the same low-level GPU programs used in game engines and creative coding tools. It gave shader authors a standard way to declare inputs, audio reactivity, and multi-pass rendering. It is now an [open standard](#) supported across a range of VJ tools.

ISF is a blank canvas. Anything you can describe mathematically can become a Visualizer, from a clean spectrum analyzer to a deep feedback loop. Typed inputs and a structured editor make it the more approachable of the two engines to write and adjust.

Finding more ISF presets

Because ISF is an open standard you do not have to build everything yourself. The [ISF website](#) and the [ISF editor](#) are good starting points, and shader collections turn up regularly on [Shadertoy](#) and in public code repositories. Drag a `.fs`, `.frag`, or `.json` file into the Visualizers editor and Visibox imports it into My Visualizers, converting it to its own format on the way in.

Editing ISF presets

The [ISF documentation](#) is the reference for the format.

Tab	What it is
Preset Defaults	The input values every Clip starts from. A Clip's own Tweak values override them.
ISF Shader	A full code editor for the GLSL fragment shader, with GLSL syntax highlighting.
Vertex	An optional vertex shader. Leave it blank for the default pass-through.
Inputs	Declare typed inputs: numbers, colors, and the audio inputs that hand waveform and frequency data to the shader.
Passes	Multi-pass rendering, where each frame can build on the last.
Info	Name, description, credits, and categories. Comments in the ISF header sync both ways, so an edit in the code updates the Info tab and back.

ISF shaders take audio through two input types:

- **audio** for the raw waveform.
- **audioFFT** for the frequency spectrum.

MilkDrop

MilkDrop was written by Ryan Geiss and released in 2001 as a Winamp plugin, where it became one of the most downloaded plugins ever made. For a generation of music fans it defined what a visualizer was: an endlessly evolving light show driven entirely by the audio.

Presets pair a simple equation language (EEL) for per-frame logic with GLSL shaders for the GPU work, which is what gives them their organic, alive quality. MilkDrop went open source, authors have been writing presets for it for over two decades, and modern implementations bring it to current hardware.

Visibox ships with a curated collection of MilkDrop presets, from classic favorites to recent community work.

Finding more MilkDrop presets

Two decades in, the archive of **.milk** presets on the web is enormous. Ryan Geiss's [Geisswerks site](#) is the original home of MilkDrop, and large community packs are easy to find. Drag **.milk** files into the Visualizers editor and Visibox imports them into My Visualizers, converting them to its own format.

Editing MilkDrop presets

Tab	What it is
Base Values	Numeric parameters for overall behavior: zoom, speed, warp, decay, and more. Use the sliders or type values.
Per-Frame	EEL equations that run every frame and can modify base values on the fly, so zoom can oscillate on a sine wave or track bass energy.

Tab	What it is
Per-Vertex	EEL equations that run for each vertex in the mesh, creating spatial variation across the screen.
Waveforms	Up to four waveform overlays drawing audio shapes. + Add Waveform adds one, up to the four.
Shapes	Up to four geometric shapes you can position, scale, and color. + Add Shape adds one, up to the four.
Shaders	GLSL code for the warp and comp (composite) passes. This is where the most dramatic effects happen.

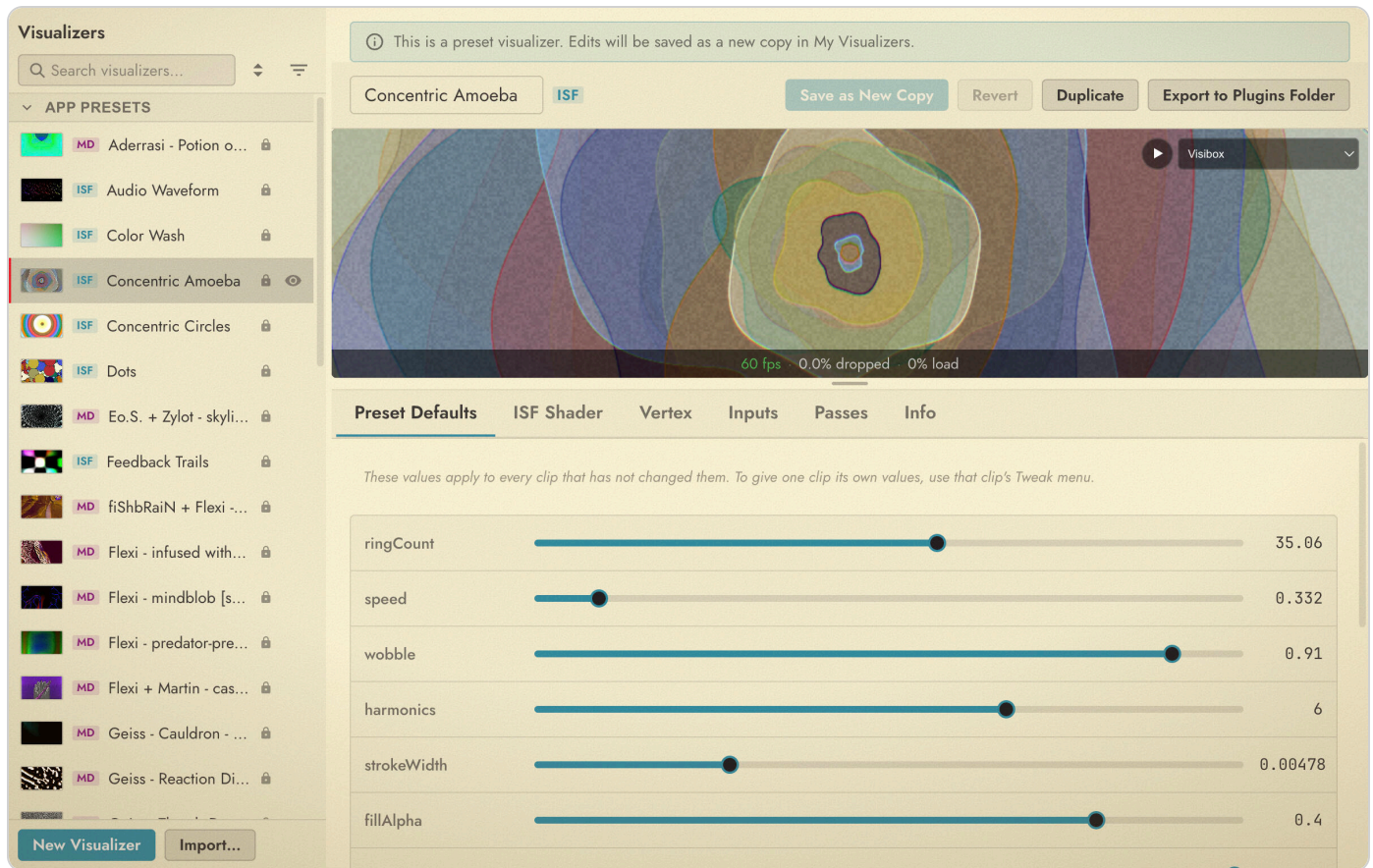
NOTE

Editing MilkDrop presets calls for some familiarity with EEL or GLSL. The [MilkDrop Preset Authoring Guide](#) is the reference.

The Visualizers editor

The Visualizers editor is a tab in the [Editor Window](#). Open it with **Visualizers...** () from the Visibox menu on Mac or the File menu on Windows.

The left panel lists every available preset. The right panel edits the selected one, with a live preview. Drag the divider to resize.



The Visualizers editor: browse presets on the left, edit and preview on the right.

Where presets come from

The list is grouped into four sections:

Section	What is in it
App Presets	The ISF and MilkDrop presets that ship with Visibox. Read-only.
Plugins	Presets loaded from your Plugins folder at startup. Read-only.
My Visualizers	Presets you created or imported. Fully editable, stored on your machine.
Project Visualizers	The copies stored inside the Project you have open. These are what actually run.

Linked and unlinked

Adding any preset to a Song puts a **linked** copy in the Project. A linked copy:

- **Stays in sync.** Update the original in My Visualizers and every Project holding a linked copy picks the change up the next time you open it.
- **Cannot be edited in place.** Unlink it first.

To unlink, right-click the preset under Project Visualizers and choose **Duplicate to My Visualizers**. That gives you an independent copy. Changes to it do not touch the original, and changes to the original do not touch it.



TIP

Keep presets linked when you want one look across several Projects. Unlink when a particular set needs its own.

Projects are self-contained

Every Project stores a complete copy of its Visualizers, linked ones included:

- **Deleting one of My Visualizers will not break a Project.** The Project keeps playing from its own copy. Visibox notes that the original source is gone.
- **Projects are portable.** Your `.vsbx` Project carries everything it needs to play back elsewhere.

Browsing

- **Search** by name.
- **Filter** by section or by visibility.
- **Sort** alphabetically or by section.
- **Hide** presets you never use with the visibility toggle on hover.

Changes to the list can be taken back while the editor tab is showing. Press `⌘Z` or use **Edit > Undo**, and the Edit menu names the step: **Undo Hide Visualizer**, **Undo Show Visualizer**, **Undo Remove Visualizer** and so on.

Audio audition

To hear how a preset responds, hover its preview and click the play button that appears. Visibox plays a test signal so you can judge the audio reactivity before putting the preset in a Project.

Performance readout

The preview carries a performance strip showing frame rate and load for the selected preset. Use it to spot presets that are too demanding for your machine, which matters most in a live show where dropped frames are visible.

Syntax highlighting

The code editors highlight GLSL, HLSL, and MilkDrop's EEL equation language, which makes complex multi-pass presets much easier to read.

Creating a Visualizer

Click **New** and choose ISF or MilkDrop. The new preset lands in My Visualizers, ready to edit.

Editing an App Preset or a Plugin is allowed, but saving writes a new copy into My Visualizers rather than overwriting the original. The editor says so before you save.

Importing and exporting

Visibox's own Visualizer format is `.vbvs`. ISF and MilkDrop files have to be *imported*, which converts them to `.vbvs`. A `.vbvs` file can be imported or installed as a plugin.

Importing

Drag any supported file into the Visualizers editor. Visibox imports it into **My Visualizers**, converting anything that is not already `.vbvs`.

Format	Extension	Description
Visibox Visualizer	<code>.vbvs</code>	Visibox's own Visualizer format.
ISF Fragment Shader	<code>.fs</code> , <code>.frag</code>	GLSL fragment shaders in ISF format.
ISF JSON	<code>.json</code>	ISF preset bundles with metadata.
MilkDrop Preset	<code>.milk</code>	Classic MilkDrop preset files.

Exporting and installing as a plugin

Right-click any preset and choose **Export to Plugins Folder** to write it out as a `.vbvs` file. `.vbvs` files work on Mac and Windows alike, so this is also how you share your work.

To install a file someone sent you, put it in your Plugins folder:

- **Mac:** `~/Library/Visibox/Plugins/`
- **Windows:** `%APPDATA%\Visibox Plugins\Plugins\`

Plugins load at startup and appear in the Plugins section of the list. The Windows folder moved in 6.0, and Visibox moves what is already there the first time it starts. See [Importing and exporting](#) for the detail.

MilkDrop texture compatibility

Some MilkDrop presets reference external texture files that classic MilkDrop loads from a `Textures/` folder next to the preset. Visibox does not load external textures, so importing one of those presets raises a **Preset Compatibility Warning** listing what is missing. The preset still imports and still runs, but anything that depended on a missing texture will not render. If a preset looks wrong after import, a missing texture is the first thing to suspect.

Overlays

An Overlay is an image that sits above everything the Output shows. A band logo in a corner all night, a lower third naming the act, a sponsor mark for one Song. Clips, Visualizers, cameras and the background all play underneath it.

NOTE

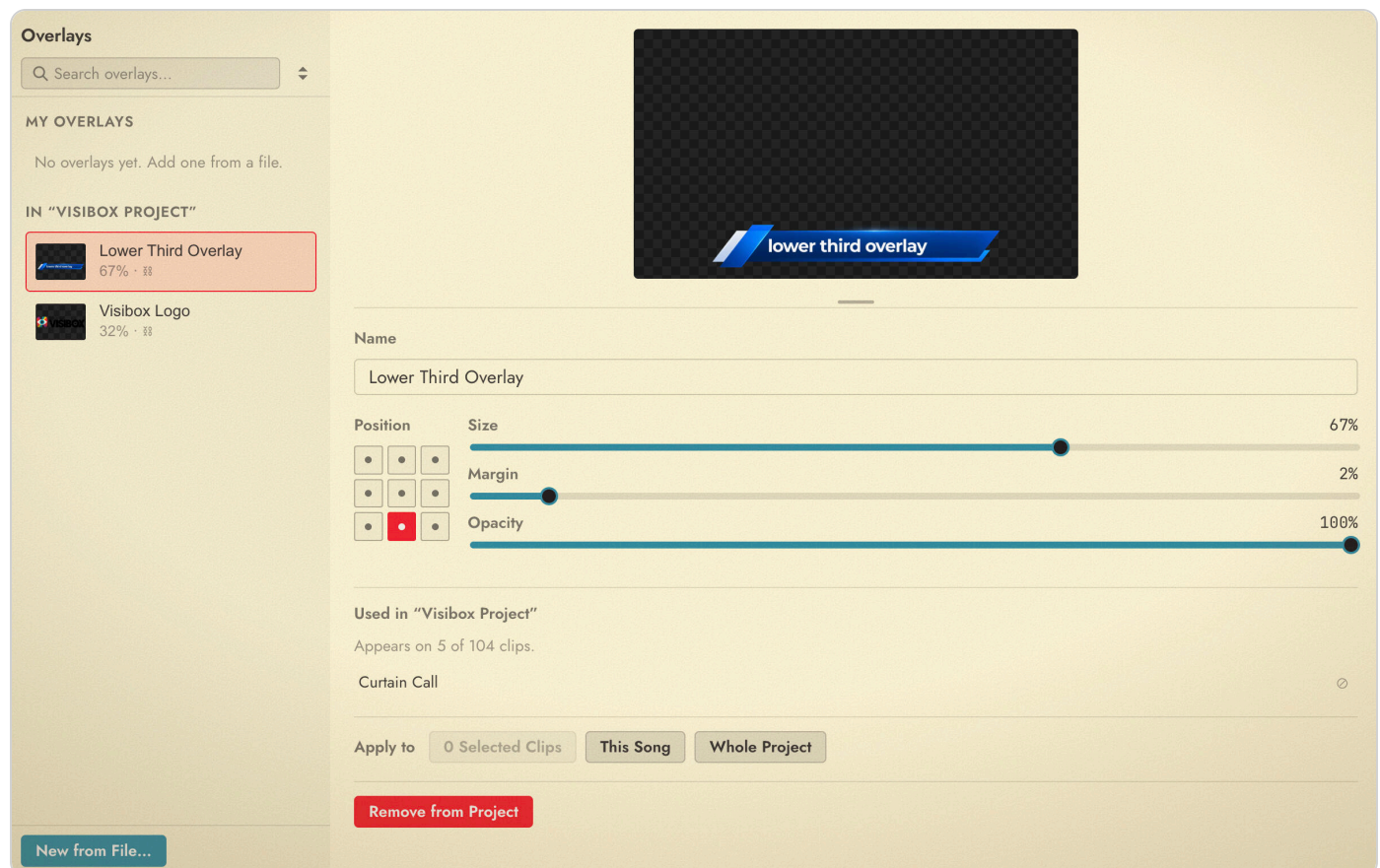
This is not the Media Overlay panel in [Manage Media](#). That panel is for reviewing and optimizing the files in a Project. Overlays are graphics that appear on the Output.

Opening the Overlays Editor

Choose **Visibox > Overlays...** to open the Overlays Editor.

Choose **File > Overlays...** to open the Overlays Editor.

You can also get there from any Overlay row in a popup menu by choosing **Edit Overlays...** at the bottom of its list.



The Overlays Editor. The library is on the left, the live preview and layout controls are on the right.

Adding an Overlay

Drag one or more image files onto the editor window, or click **New from File...** and pick an image.

The left side of the window holds two lists. **My Overlays** is your own library, which follows you between Projects. **In This Project** holds the Overlays this Project actually contains. Applying a library Overlay to a Project copies its image into the Project folder, so the Project travels with its graphics.

Both lists can be searched and sorted by name or by size.

Placing an Overlay

Select an Overlay to edit it. The preview above the controls is framed at the shape your Output is set to, so what you position against is what the audience gets.

Control	What it does
Name	What the Overlay is called in the library and in every Overlay menu.
Position	A three by three grid of anchor points: the corners, the edge centers, and the middle.
Size	Overlay width as a percentage of the Output width, from 1% to 100%. Height follows the image's own shape.
Margin	Distance from the anchored edges, as a percentage of the Output's shorter side. Up to 50%.
Opacity	How solid the Overlay is, from invisible to fully opaque.

Because size and margin are percentages rather than pixels, an Overlay keeps its proportions when the Output resolution changes. A logo set up on a 1080p rehearsal screen lands in the same place on a 4K wall.

Layout belongs to the Overlay itself, not to the Clip it appears over. Change the margin once and every place that Overlay is used moves with it.

Assigning an Overlay

Overlays are assigned at three levels, each with its own **Overlay** row:

- **Project** in the Project menu, for something that should be up all night.
- **Song** in the [Song Editor Panel](#), for a lower third that belongs to one number.
- **Clip** in the [Clip Editor Panel](#), for a mark tied to one piece of media.

The fastest way to assign the current selection is the **Apply to** bar at the bottom of the Overlays Editor, which offers the selected Clips, the current Song, or the whole Project.

Each Overlay row offers **Inherit**, **None**, and your Overlay library. The row reports what it currently resolves to, so an inheriting Song reads as “Inherit (Tour Logo, from Project)” rather than a bare “Inherit”.

How the levels stack

The three levels add up rather than replace each other. A Project logo and a Song lower third both appear.

- **Inherit** is the default. The level passes through whatever the levels above it produced.
- Choosing Overlays at a level **adds** them on top of what was inherited.
- **None** suppresses everything inherited so far. A Song set to None hides the Project logo for that Song, and a Clip set to None hides both.

Up to eight Overlays can be stacked at each level. They paint in order, so an Overlay added lower in the hierarchy sits in front of one added higher up.

Removing an Overlay

The button at the bottom of the editor changes with what is selected. On a Project Overlay it reads **Remove from Project** and takes the Overlay out of this Project, leaving your library untouched. On a library Overlay it reads **Delete Overlay** and takes it out of the library.

Adding, editing and deleting a library Overlay can all be taken back. Press  **Z** while the editor is showing, or use **Edit > Undo**, and the Edit menu names the step: **Undo Add Overlay**, **Undo Update Overlay** and **Undo Remove Overlay**. A deleted Overlay keeps its image until then, and Visibox clears out images no Overlay uses the next time it starts.

The usage panel above that button lists every Clip, Song and Project the selected Overlay appears in, so you can see what a removal costs before you make it. Song and Clip entries can be revealed in the Controller, and any entry can be unassigned on the spot.

When an image goes missing

Overlay images live in the Project folder like any other media. If a file is moved or deleted, Visibox says which Overlay lost its image and keeps the entry so the layout survives. Add the image again to restore it.

Lyrics

Lyrics puts a chord and lyric sheet on its own screen, facing the stage, while Visibox drives the visuals out front. Paste the chart you already have, pick a size you can read from where you stand, and Visibox follows the Song as you play it.

It reads the chart formats musicians actually keep songs in. You do not convert anything.

Getting a Sheet In

Select a Song, then choose **Song > Edit Song Lyrics... (Beta)**. The Song's right-click menu has the same row. Paste your chart into the pane and Visibox works out what it is.

Turn on Settings > General > Enable Beta Features to see this.

Nothing to configure and no format to pick. Visibox reads:

Format	What it looks like
ChordPro	Chords in brackets inside the words: [C]Hello dark[Am]ness
OnSong	ChordPro with OnSong's header lines. Reads as ChordPro.
Chords over lyrics	A line of chords sitting above a line of words. The way most tab sites print a song.
OpenSong	Chord lines starting with a dot, lyric lines starting with a space.
Nashville	Chords written as numbers: 1 4 5, 6m, b7.
iReal Pro	An irealb:// link pasted straight in.
Lyrics only	Words with no chords at all.

The status bar at the bottom of the editor names the format it detected, so you can see at a glance that your paste came through as you expected.

Visibox also reads bar charts, fingerpicking patterns, and ASCII tab, wherever they turn up in the sheet.

Song information

Visibox picks up a line like Title: Wichita Lineman or a ChordPro directive as song information rather than lyrics:

```
{title: Wichita Lineman}
{artist: Jimmy Webb}
{key: F}
{capo: 2}
```

The directives Visibox reads are `title`, `subtitle`, `artist`, `composer`, `album`, `year`, `key`, `capo`, `tempo`, `time`, `duration`, and `tuning`. Visibox leaves anything else alone and notes it as a warning, so a sheet from another app never breaks.

Visibox understands plain lines too: `Capo: 2`, `Capo II`, and `Capo: none`. The line has to start with the word `Capo`. After `Capo:`, the words `none`, `off`, and `no` all mean no capo.

Sections

Mark a section any of these ways:

```
[Verse 1]
Chorus:
{start_of_chorus} ... {end_of_chorus}
---
```

Three hyphens on their own line start a new section without labeling it.

Square brackets take any label you like. The plain `Chorus:` form is the narrower one: it works for the section names Visibox knows, which are Intro, Verse, Chorus, Pre-Chorus, Bridge, Outro, Ending, Coda, Interlude, Instrumental, Break, Solo, Tag, Refrain, and Hook. Put a name of your own in square brackets instead.

A `#` at the start of a line is a note to yourself. It never appears on the screen.

Chord diagrams you define yourself

If a chord in your sheet has a voicing you want shown, define it:

```
{define: Cm base-fret 8 frets 8 10 10 8 8 8}
```

There is a shorter form as well, for when you are typing from a chord box on paper:

```
Cm    8-10-10-8-8-8
```

Frets run from the low string to the high one. `x` is a muted string and `0` is an open one. A definition in the sheet wins over the built-in shape.

Timing a Sheet

An untimed sheet scrolls at a steady rate. A timed sheet follows the Song, which is better, and stamping the times takes one pass.

Put a time marker at the start of a line and Visibox knows when that line arrives:

Marker	Means
<code>[0:12]</code>	Twelve seconds into the Song.

Marker	Means
[1:05.5]	The same, to a fraction of a second.
[75]	Plain seconds from the start.

A marker has to open the line. A marker alone on its own line hands its time to the line below it. That is how you time a chords-over-words pair: put the marker on a line of its own above the pair, and it stays out of the chord columns.

Tap timing

Typing markers by hand is slow. Tap them in instead:

1. Open the Song's lyrics in the editor.
2. Turn on **Tap timing** in the status bar at the bottom.
3. Trigger the Song and let it play.
4. Press  as each line arrives. Visibox stamps the current time onto the line at the cursor and moves down to the next one.

The readout beside the checkbox shows where the Song is, so you know the timing is live before you start tapping. One press is one marker, so holding the keys down does not stamp a run of lines.

The markers are ordinary text in the sheet. Fix a mistimed line by editing its number.

The Lyrics Window

Choose **Window > Lyrics Window (Beta)** to open it. It follows whichever project and Song are active, so it keeps up as you move through the set without being touched.

Turn on Settings > General > Enable Beta Features to see this.

The screenshot shows a dark-themed interface for a lyrics window. At the top, a controls bar is visible with a pin icon on the left and a close icon on the right. The lyrics are displayed in a white font on a dark background. Chords are shown in a smaller font above the lyrics. The lyrics are: "Paste the chart you use already", "Let the pages find their way", "One small tap to mark the timing", "And it knows the song each day", and "Keep your place, stay in the moment". The word "CHORUS" is written in red above the last line. The chords are: Bm7, D, Em, G A, Bm7, D, G F A, Bm7, D, E.

The Lyrics window.

The controls bar sits at the top and slides away after a few seconds of not being used, leaving the sheet alone on the screen. Move the mouse to bring it back, or click the pin to keep it up while you set things.

View modes

Mode	What you get
Teleprompter	The whole sheet scrolling continuously, with the current line in the middle.
Stage	The current line large, with the next two lines below it, dimmed.

Teleprompter is the default and the one most players want. **Stage** is for a small screen and a big room.

Scrolling

Scroll speed is set in lines per minute, from 0 to 60. At 0 it reads **Off** and the sheet stays where you put it.

If the sheet carries time markers, the control reads **Timed** and turns itself off, because Visibox is following the Song instead of guessing a rate. Scroll the sheet by hand at any point and it picks up again from where you left it rather than snapping back.

Chords and diagrams

Control	Options
Chords / Lyrics	Show the chords, or hide them and read the words only.
Chord display	Chord names , Diagrams (first use) , or Diagrams (every chord) .
Instrument	Guitar , Ukulele , or Piano .

Diagrams (first use) is the useful middle setting: a diagram the first time a chord appears, its name after that. Guitar and ukulele get fretboard boxes, piano gets a small keyboard with the notes filled in.

Visibox has open-position shapes for the common chords and works out a barre shape for anything else, so a chord you have never seen still gets a diagram.

If your sheet contains tab, a **Tab** control appears with two settings: **Tab: notation** draws it as a clean fretboard diagram, **Tab: text** leaves it as the monospaced ASCII you pasted.

Transpose and capo

Transpose moves the chart a semitone at a time. The readout shows **+0** at rest, then **#2** or **b3** as you move it. Chord shapes and any diagrams follow, and Transpose leaves the chord quality alone: a **Cmaj7** becomes a **Dmaj7**, not a **D**.

Transpose is per-Song and deliberately temporary. It resets when you move to another Song, so a chart you dropped a tone for one singer does not follow you into the next number.

If the sheet declares a capo, a second control appears with two settings:

- **Capo 2** (or whatever fret the sheet names) shows the chords as printed. This is what you fret with the capo on.
- **Concert** shows the chords as they actually sound, so a piano or horn player reading over your shoulder is in the right key.

Text size and theme

Text size runs from 50% to 300%. **⌘+** and **⌘-** step it, and **⌘0** puts it back to 100%. Set it once for the screen you are standing at and it stays.

Light and **Dark** switch the sheet's theme. Dark is the default, and it is the right one for a dark stage.

An **Edit** button in the controls bar opens the current Song's sheet in the editor. Fix a wrong chord between songs without leaving the window.

Putting Lyrics on Their Own Screen

Lyrics get a screen of their own, separate from the output your audience sees and separate from the Controller.

1. Open **Settings > Canvas**.
2. Select the screen you want in the display arranger.
3. Set its **Role** to **Lyrics**.

A screen assigned to Lyrics is not an output target and is never picked for the Controller, so nothing else can land on it by accident. Once you make the assignment, **Present Full Screen** drives the Lyrics window full screen onto that display along with everything else.

If the screen is not plugged in, or the Lyrics window is closed, Present Full Screen skips it. A missing lyrics monitor never stops a show.

 NOTE

Lyrics show on their own display, not on the main output. If you want words in front of the audience, that is an Overlay or a Clip, not this.

Tips

 TIP

Tap the timings once at soundcheck and the sheet follows the Song every night after. It is a few minutes per song and it is the difference between reading and scrolling.

 TIP

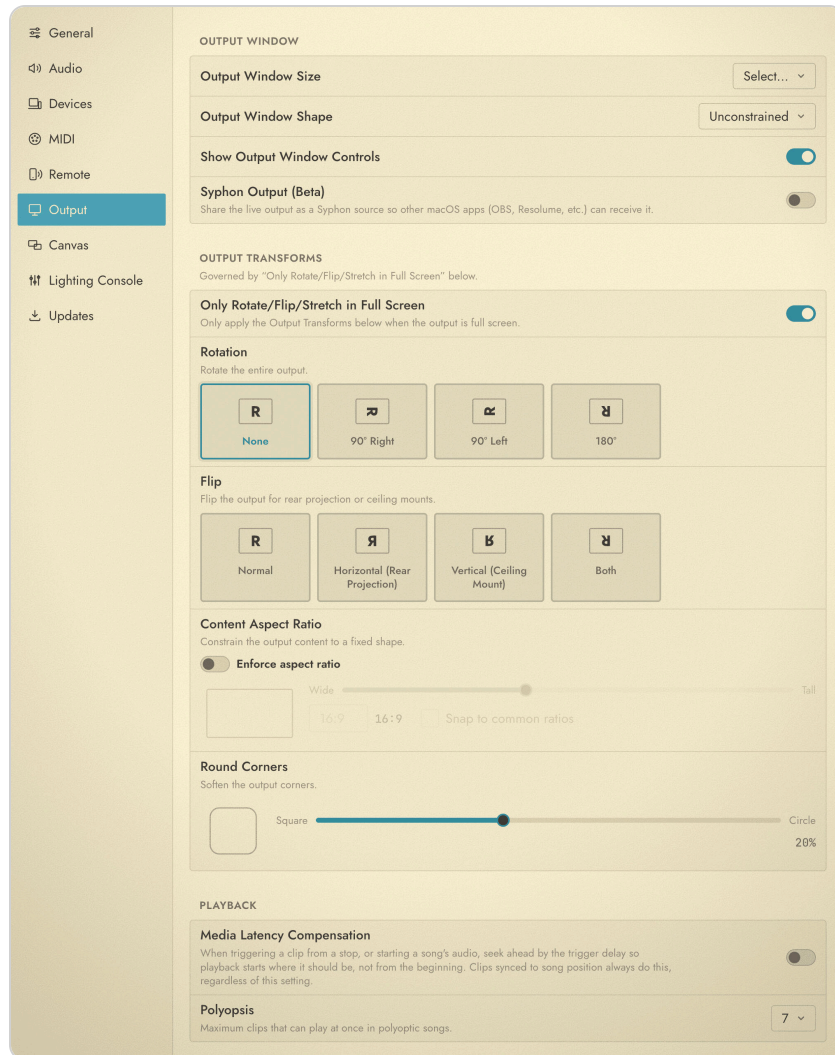
Set **Chord display** to **Diagrams (first use)** for a song you are still learning and **Chord names** for one you know. It is the fastest way to keep the sheet readable at the size you need.

See Also

- [Song Timeline](#) for the other tab in the same editor window.
- [Adding Songs](#) for attaching the Song audio that timed lyrics follow.

Settings

The Settings window holds everything you configure once and then forget about: audio hardware, cameras, MIDI, the shape of the Output, which screen shows what, and how Visibox behaves at launch.



Settings. Pick a category on the left, and its controls fill the panel on the right.

Opening Settings

Choose **Visibox > Settings...**, or press  , .

Choose **File > Settings...**, or press  , .

Closing the window puts it away rather than shutting it down, so it reopens on the category you were last using.

Some menu items open Settings directly on the category they belong to. **MIDI Map...** lands on MIDI, **Arrange Displays...** lands on Canvas, and **Pair with Remote...** lands on Remote.

The nine categories

They are listed here in the order the sidebar shows them. General is first because it is where you look when you do not yet know which category you need, and it is where the window opens the first time.

Category	What it covers
General	Language, launch behavior, clock and title options, and confirmations
Audio	Output device, channel count, master volume and mute, and the audio input that feeds Visualizers and Effects
Devices	Cameras and their capture settings, and the Stream Deck plugin
MIDI	Which controllers Visibox listens to, and the MIDI Map: which note or knob on each one drives which action
Remote	Pairing the phones, tablets, and AI tools that drive Visibox over the network, and the OSC port
Output	The size and shape of the Output, its transforms, and how many Clips can play at once
Canvas	Which screen shows the Output, the Controller, and the Lyrics, plus video wall, NDI, and DeckLink outputs
Lighting Console	Letting a lighting desk drive Visibox over sACN or Art-Net, and the fixture profile it patches
Updates	The version you are running, and checking for a newer one

What moved in 6.0

If you are coming from Visibox 5, five things are somewhere new.

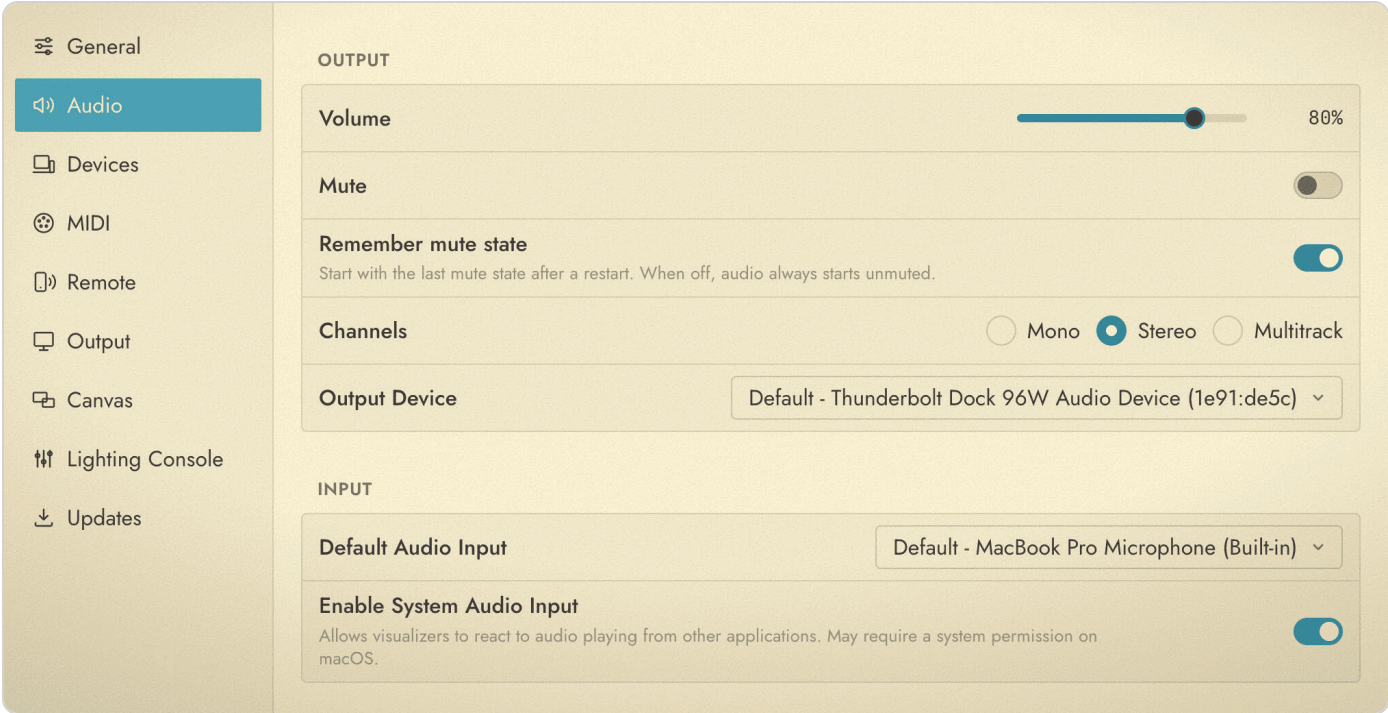
- **The MIDI Map is a category, not a window.** It opens inside Settings, and the list of MIDI inputs sits above it, so turning a controller on and mapping it happen in one place. The **MIDI Map...** menu item still exists and still gets you there. See [MIDI Settings](#).
- **Remote pairing is a category, not a window.** Visibox 5 had a Remote Pairing window under the Window menu. It is now the Remote category, and **Pair with Remote...** opens Settings on it. See [Remote Pairing](#).
- **Stream Deck Mode is gone.** The Elgato plugin is the only way Visibox talks to a Stream Deck now. Install it from [Devices](#), and see [Elgato Stream Deck](#) for the rest.
- **The Project menu's output transforms moved here.** Rotate, Flip, Enforce Aspect Ratio, Round Corners, and Only Rotate/Flip/Stretch in Full Screen are in [Output](#). Video Wall Controller Compensation is in [Canvas](#).
- **Updates has its own category.** It shows the version you are running and when it was last checked. See [Updates](#).

Project title, Uppercase Titles, and Default Fit Mode are not in this window at all. They belong to the individual Project, and live in the [Project Settings](#) window.

Audio Settings

The Audio tab is where you tell Visibox which device to play out of, how many channels to use, and which input feeds audio-reactive Visualizers and Effects.

This page is the control reference. For how multichannel playback, stem files, metering, and ASIO output actually work, see [Audio](#).



Settings > Audio, split into an Output section and an Input section.

Output

Setting	What it does
Volume	Master output level, shown as a percentage. Moving the slider while muted unmutes.
Mute	Silences all output while holding the volume slider in place, so you can see where the level returns to when you unmute.
Remember mute state	When on, Visibox starts with the mute state you left it in. When off, audio always starts unmuted.
Channels	How Visibox maps audio onto your interface: Mono , Stereo , or Multitrack . The same three modes appear on the Control Bar, where Multitrack reads “Multichannel”.
Output Device	Which audio device Visibox plays through. The list holds the output devices your operating system reports, and the first entry follows the output chosen in your system settings. If Visibox finds none, the row reads “No audio output devices found”.

Volume and Mute are also on the Control Bar, and both have keyboard shortcuts.

 for Volume Up,  for Volume Down, and  for Mute.

 for Volume Up,  for Volume Down, and  for Mute.



TIP

If you have an external audio interface for the show, set Visibox to that interface and leave your system output on the built-in speakers. System notifications and alerts then stay off the front-of-house feed.

ASIO interfaces on Windows

ASIO interfaces are picked from the Control Bar rather than from the Output Device list above. Click the output level meter at the right of the Control Bar to open a menu listing every output device. Your ASIO interfaces appear below the operating system's own devices. Turn on Settings > General > Enable Beta Features to see them. See [Audio](#) for what changes while an ASIO device is active.

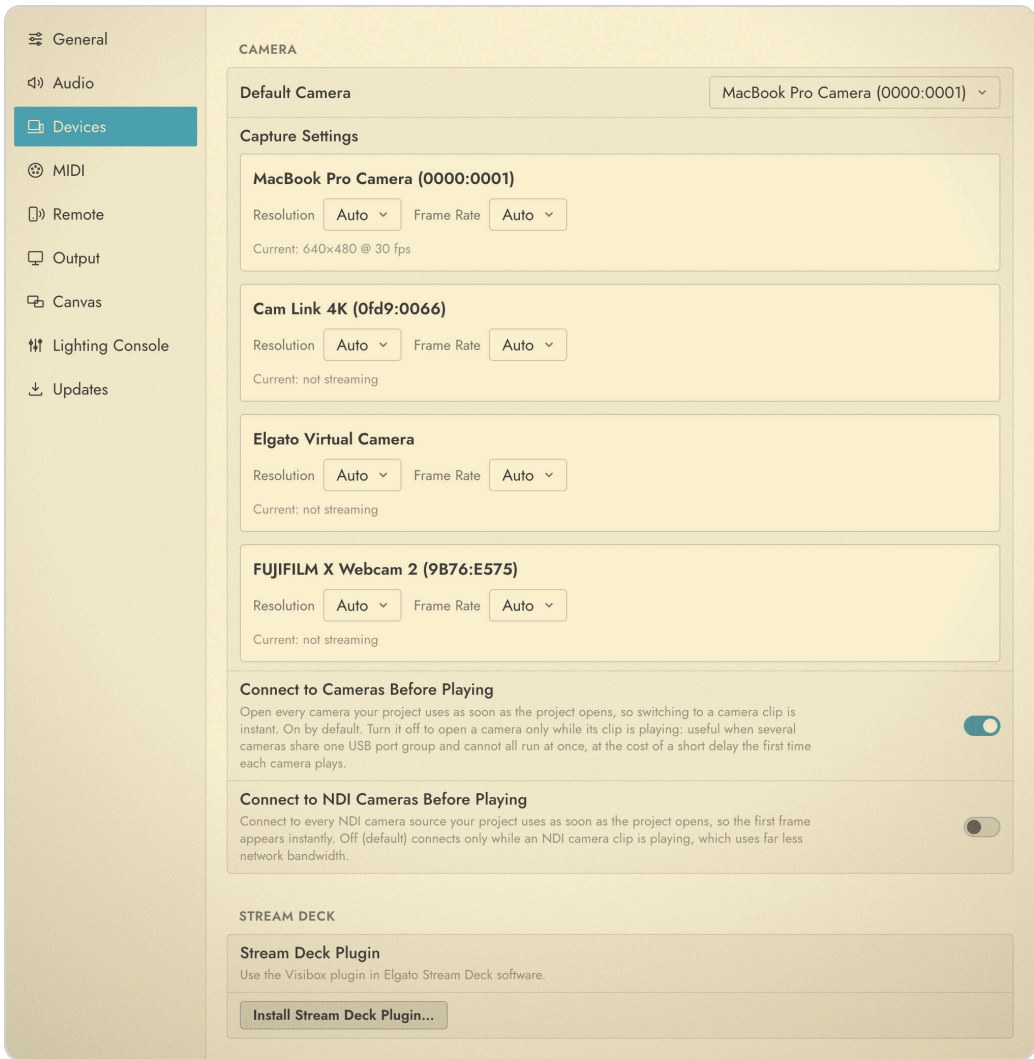
Input

Setting	What it does
Default Audio Input	The input device Visualizers and audio-reactive Effects listen to when they are set to follow a live input. If Visibox finds none, the row reads “No audio inputs found”.
Enable System Audio Input	Lets Visualizers and Effects react to audio playing from other applications on your computer as well as audio inside Visibox. On macOS this needs a system permission.

See [Visualizers](#) and [Effects](#) for what each one does with an audio source.

Devices Settings

Cameras and your Stream Deck. MIDI controllers are turned on and mapped in [MIDI](#).



Devices settings. Camera capture and the Stream Deck plugin.

Camera

Default Camera picks which camera or capture device new Camera Clips use. Every connected USB camera and HDMI capture adapter appears in the list. If nothing is connected, the section reads “No cameras found”.

Under it, **Capture Settings** gives each connected camera its own two controls.

Control	What it does
Resolution	4K UHD (3840×2160), 1440p, 1080p, 720p or 480p, or Auto to let the camera decide.
Frame Rate	60, 30 or 24 fps, or Auto .

Visibox asks the camera what it can do and only offers the presets it actually supports, so a 1080p webcam does not list 4K. Next to each camera is a **Current:** readout of what the camera is really sending, as width × height @ frame rate. It reads “not streaming” until something is using that camera.

Auto is not the same as a preset. Auto removes the override entirely and lets the camera pick, which is usually what you want unless a capture device is negotiating a resolution you do not like.

Two switches decide when Visibox connects to cameras:

Setting	What it does
Connect to Cameras Before Playing	On by default. Opens every camera the Project uses as soon as the Project opens, so switching to a Camera Clip is instant. Turn it off to open a camera only while its Clip is playing. That costs a short delay the first time each camera plays, but it helps when several cameras share one USB port group and cannot all run at once.
Connect to NDI Cameras Before Playing	Off by default. Connects to every NDI camera source the Project uses as soon as the Project opens, so the first frame appears instantly. Left off, Visibox connects only while an NDI Camera Clip is playing, which uses far less network bandwidth.

See [Connecting Cameras & Video Inputs](#) for getting a camera into a Song in the first place.

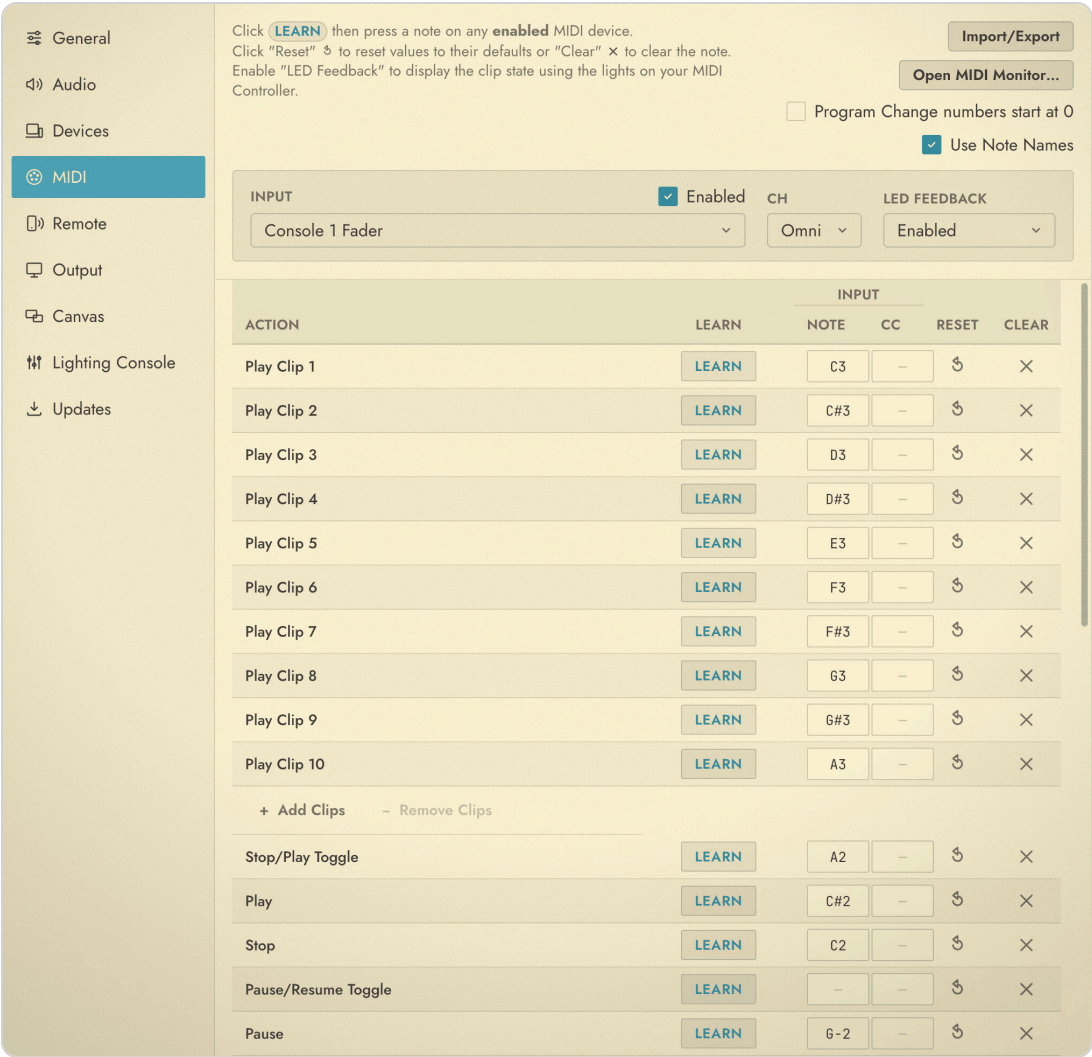
Stream Deck

Visibox drives an Elgato Stream Deck through the official Visibox plugin, running inside Elgato’s own Stream Deck software. See [Elgato Stream Deck](#) for how the plugin works, its actions, and what changed from Visibox 5’s built-in mode.

Click **Install Stream Deck Plugin...** to install it. Elgato’s software has to be installed first, since it is what opens the plugin file. If the install cannot go through, Visibox says why underneath the button.

MIDI Settings

This is the MIDI Map: a table of everything Visibox can be told to do, and the note or knob on your controller that tells it. In Visibox 5 this was a separate window. It is now a category in Settings, and the **MIDI Map...** menu item opens it here.



The MIDI Map. Pick a device at the top, then assign notes and CCs to actions in the table.

Maps are stored on your computer, not in the Project, so a map you build once follows you from show to show.

Choosing a device

The bar above the table decides which controller you are working on, and whether Visibox is listening to it.

Control	What it does
Input	The device this map belongs to. Every device gets its own map. Default is the fallback map, used for any device that has no map of its own.

Control	What it does
Enabled	Beside Input. Checked, Visibox listens to this device. A device you uncheck stays in the list marked (off) , so you can map a controller before you turn it on. The Default map has no checkbox.
Ch	The MIDI channel to listen on, 1 through 16, or Omni for all of them.
LED Feedback	Enabled or Disabled . When enabled, Visibox lights the pads on this controller to show what is playing. See MIDI Feedback .

New devices arrive enabled, so a controller you plug in works without a visit here. See [Using MIDI](#) for when you would turn one off.

Unplug the device you are editing and Visibox falls back to the Default map so you are never editing something that is not there.

Open MIDI Monitor..., at the top of the category, opens the [MIDI Monitor](#) window, which shows the raw messages arriving from your controller.

Use Note Names at the top of the table switches the Note column between numbers (0 to 127) and note names such as C4 or D#5. It changes the display only, not the mapping.

Program Change numbers start at 0, beside it, switches how Visibox numbers a Song's Program Identifier: 1 to 128 when it is off, 0 to 127 when it is on. Match it to whatever your controller prints on its own display. It changes the numbering you see, not which message selects which Song. See [Selecting Songs via MIDI Program Change](#).

The table

Each row is one action, with a **Note** field and a **CC** field. Assign either, or both. The row also carries a **Learn** button and, at its right edge, **Reset** and **Clear**.

The rows run in this order.

Rows	Notes
Play Clip 1 through Play Clip 10	Trigger a Clip by its position in the Song.
Add Clips / Remove Clips	Buttons, not actions. Add Clips appends five more Play Clip rows for Songs with more than ten Clips; Remove Clips takes them back.
Stop/Play Toggle, Play, Stop, Pause/Resume Toggle, Pause, Resume	Transport.
Previous Song, Next Song, Previous Song - No Stop, Next Song - No Stop	Song navigation. The No Stop pair moves on without stopping what is playing.

Rows	Notes
Previous Clip, Next Clip, Previous Clip - Unconstrained, Next Clip - Unconstrained	Clip navigation. The Unconstrained pair keeps going past the end of the current Song.
Output Volume	Master level. CC only, since it needs a continuous value.
Control A through Control H	Real-Time Control. CC only. See below .
Song Select	Scrubs the whole Song list with one knob. CC only.
Select Song via Program Change	A dropdown, not a note or CC. See Selecting Songs via MIDI Program Change .

A note or CC number can only drive one action at a time. Assign one that is already in use and Visibox clears it from the row that had it, so you never end up with two actions fighting over the same pad.

Learn

Typing note numbers works, but pressing the pad is faster.

1. Click **Learn** on the row you want to map. The row highlights and reads “Listening...”.
2. Press the pad, or turn the knob, on your controller.
3. Visibox fills in the number.
4. After a note, Learn moves down to the next row on its own, so you can walk a whole grid of pads in one pass. After a CC it stops, since knobs are usually mapped one at a time.

While a row is armed, the keyboard helps:

-  and  move to the previous or next row.
-  or  clears the current row.
-  stops learning.

Two things Learn deliberately ignores. CC numbers 120 to 127 are channel-mode messages, which a DAW fires off by itself when the transport starts, so Learn stays armed and waits for a real knob. And leaving the MIDI category, or putting the Settings window away, disarms the row: a hidden window listening for MIDI would otherwise swallow notes you meant for the show.

Assigning a Control

Controls A through H are the named Real-Time Control slots you point at a Clip or Effect parameter. Binding one to a knob is the same procedure as any other row, with one difference: Control rows take CC only, because a Control needs a value to sweep, not a press.

1. Scroll to the **Control A** row, or whichever letter you want.
2. Click its **Learn** button.

3. Turn the knob or move the fader you want to own that Control.
4. The CC number appears in the row, and Learn stops.

That is the whole binding. Which parameter the Control actually moves is decided in the Project Settings window, on its Control Bus tab.

Reset, Clear, and starting over

Reset puts one row back to its default assignment. **Clear** empties it.

What “default” means depends on which map you have open. Editing a device’s own map, Reset pulls the value from your Default map. Editing the Default map itself, Reset pulls from the assignments Visibox ships with.

At the bottom of the table, under **Global actions**, **Reset all** and **Clear all** do the same thing to every row at once.

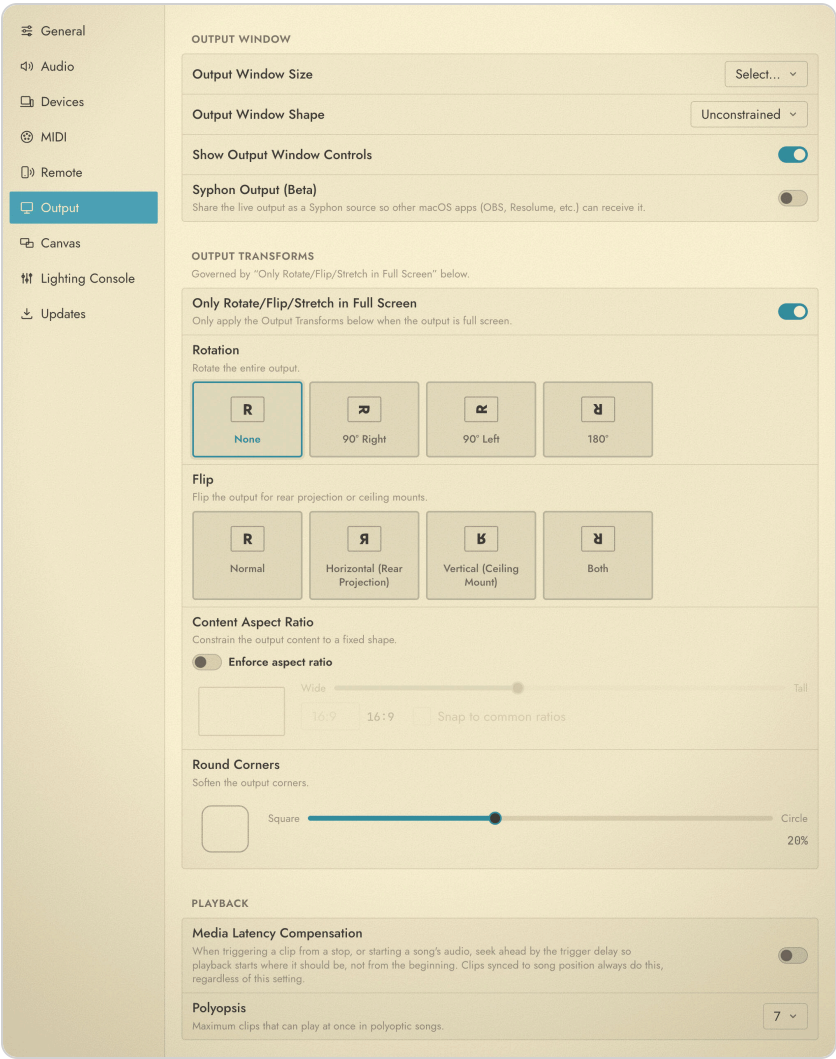
Import and Export

The **Import/Export** button at the top saves the current map to a `.vsbmm` file, or loads one back in. Use it to back a map up before a big edit, to move a controller’s map to another machine, or to hand a working map to someone with the same controller.

Output Settings

The Output tab shapes the picture: how big it is, what shape it is, which way up it is, and how many Clips can be on it at once. It applies to Visibox as a whole, not to one Project, so a setting made here survives switching Projects.

Where the picture goes after it is made (extra displays, video walls, NDI, DeckLink, recording) is in [Canvas Settings](#).



Settings > Output.

Output window

Setting	What it does
Output Window Size	Sets the size of the Output window on screen when it is not full screen. Pick a preset from VGA up to 8K, or choose Custom, type a width and height, and click Apply. See the note below.

Setting	What it does
Output Window Shape	Constrains the Output window to a fixed shape: Unconstrained, 1:1, 4:3, 16:9, 16:10, or a custom ratio you type.
Show Output Window Controls	Shows on-screen controls on the Output window. Useful while setting up, and worth turning off before an audience sees the window.

Output Window Size sets how large the window is, and nothing else. A windowed Output renders at a fixed size, 1920 along its longest edge, shaped by the Project's aspect ratio when one is set. The picture is then fitted into whatever window you chose. It used to be rendered at the window's own size. That meant a small window quietly starved a recording, or an NDI, DeckLink, Syphon or Spout target, of the pixels it had asked for.

Fitted means the picture fills the window and any overhang is cropped. To see the whole canvas with bars instead, turn on [Letterbox Output window](#), which is also on the Output window's right-click menu.

Full screen is unaffected: the Output renders at the display's own resolution, as it always did.

Syphon Output shares the live output as a Syphon source, so other macOS applications such as OBS and Resolume can receive it. See [Output, Canvas, and Displays](#).

Turn on Settings > General > Enable Beta Features to see this.

Spout Output shares the live output as a Spout source, so other Windows applications such as OBS and Resolume can receive it. See [Output, Canvas, and Displays](#).

Turn on Settings > General > Enable Beta Features to see this.

Output Transforms

These four controls adapt the output to the room: a projector hung upside down, one firing through the back of a screen, a screen that is the wrong shape, a display you want to soften the edges of.

Only Rotate/Flip/Stretch in Full Screen decides when they apply. Leave it on and the transforms only take effect once the output is full screen, which keeps the Output window the right way up while you build the show. Turn it off to see the transforms in the windowed output too.

Rotation

Rotates the entire output: None, 90° Right, 90° Left, or 180°.

Use 180° for a ceiling-mounted projector hung upside down, and 90° for a vertical banner screen. Your projector or your operating system's display settings may offer rotation of their own, which is usually the tidier place to do it if the option is there.

Flip

Mirrors the output: Normal, Horizontal (Rear Projection), Vertical (Ceiling Mount), or Both.

Horizontal is for rear projection, where the audience sees the picture through the screen and would otherwise read it backwards. Vertical is for a ceiling mount that flips the image the other way.

Content Aspect Ratio

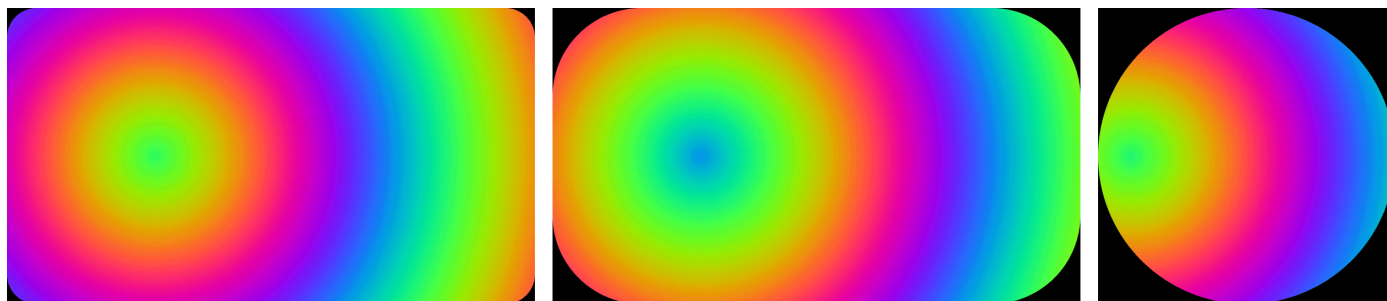
Turn on **Enforce aspect ratio** to hold your content at a fixed shape regardless of the display's shape, adding bars to make up the difference.

The slider runs from wide to tall, with 1:1 in the middle, and the readout names the ratio you are on. Turn on **Snap to common ratios** to have the slider settle on the standard shapes instead of moving freely.

This is worth setting when you do not know what you will be projecting onto. Your content then reads the same on a 16:9 projector, a 16:10 one, and whatever the venue actually owns.

Round Corners

Softens the output's corners, from Square through to Circle.



The same Output at 10%, at 30%, and at Circle with a 1:1 content aspect ratio.



TIP

Round displays

Set Round Corners to Circle and enforce a 1:1 content aspect ratio and you get a circular display area, whatever shape the screen behind it is.

Playback

Media Latency Compensation

Triggering a Clip is not instant, and neither is starting a Song's backing track. Whether the trigger is a MIDI note, a Stream Deck button or an API call, Visibox needs a moment to get the Clip ready, and that moment is longest the first time a Clip plays. The Clips in a Project's first Song are made ready as the Project opens, so this setting is about the rest.

Off (the default) waits until the Clip is ready and plays it from the very beginning. You get every frame and every sample, and the Clip lands slightly late.

On starts the Clip in sync with the trigger by giving up the moment it spent getting ready, at the cost of the first few milliseconds of content. That trim is hard to notice on steady footage, but it can clip the front off a percussive audio track, which may be the whole reason you were triggering on the beat.

Clips set to **Sync to Song Position** stay locked to the Song's timeline on their own, and this setting does not affect them.

Polyopsis

Polyopsis is the ceiling on how many Clips can play at once in polyoptic Songs, from 2 to 8.

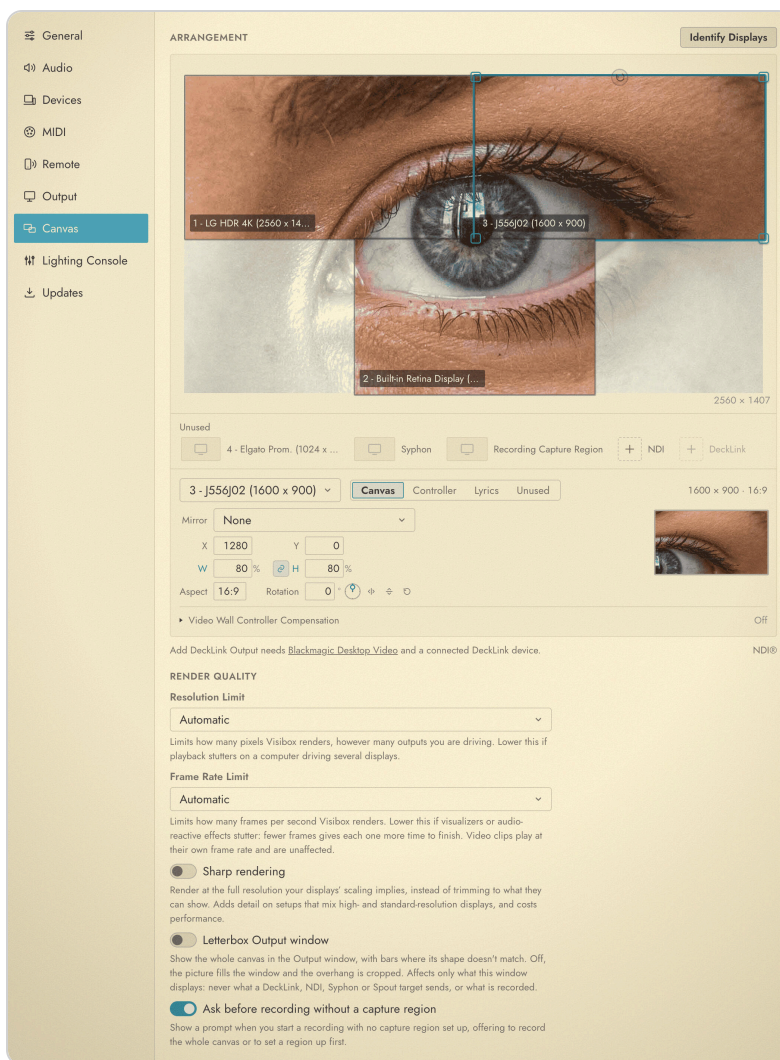
A Song's Poly Mode decides how simultaneous Clips share the stage. Polyopsis decides how many of them there can be before the oldest one gives way. Higher numbers ask more of the machine, so set it to what your show needs rather than to the maximum.

Canvas Settings

The Canvas tab is the map of where your output goes. Every screen, network stream, capture card and screen recording that takes a piece of the picture is arranged here.

Open it from Settings > Canvas, or from **Window > Arrange Displays...**, which opens the same tab. The three separate Output Display, Controller Display and Identify Displays rows that used to sit in the Window menu are all replaced by this one tab.

For what the arrangement means and how the pieces fit together, see [Output, Canvas, and Displays](#).



Settings > Canvas.

The Arrangement

The arrangement is a scale map of your output. Each rectangle on it is one destination, and the part of the picture that rectangle covers is what that destination carries. Sample content is drawn across it so you can see how the picture divides between them.

Anything not currently carrying output waits in the tray beside it, grouped under **Controller**, **Lyrics**, and **Unused**.



The arranger. Drag a display out of the Unused tray to put it to work.

- **To turn a display on**, drag it out of the Unused tray and drop it on the canvas, or select it and set its Role to Canvas.
- **To move a destination**, drag its rectangle. The output follows as you drag.
- **To inspect or edit one**, click it. The Transform panel fills in with its settings.

Smaller rectangles stack on top of larger ones, so a laptop screen sitting inside a projector’s rectangle stays clickable.

Roles

A display does one job at a time, chosen from the **Role** control at the top of the Transform panel:

Role	What the display does
Canvas	Carries a piece of the output. The display takes a rectangle on the canvas.

Role	What the display does
Controller	Shows the Controller window. The display leaves the canvas.
Lyrics	Shows the Lyrics window. The display leaves the canvas.
Unused	Shows nothing from Visibox, and waits in the Unused tray.

One display can be the Controller and one can be the Lyrics display. Choosing a new role releases the old one, so making a display the Controller takes it off the canvas, and making it Unused hands the Controller window back to wherever it was.

NDI, DeckLink, Syphon, Spout, and the recording region are not screens and cannot show a window, so they offer Canvas and Unused only.

The Transform panel

With nothing selected the panel reads “Select a display to inspect”. Select a rectangle and you get:

Control	What it does
Role	What this display is used for. See Roles .
Display	Which physical screen this rectangle drives, and its native resolution.
Position	X and Y on the canvas, for placing a rectangle exactly rather than by eye.
Width and Height	The size of the rectangle, with its resulting aspect shown alongside.
Scale	Scales the rectangle. Link width and height keeps its shape while you do.
Rotation	Rotates the rectangle on the canvas. You can also drag the rotate handle on the rectangle itself.
Flip H and Flip V	Flips what this destination shows, horizontally, vertically, or both.
Mirror	What this destination shows instead of its own slice of the canvas. See Mirror .
Mirror Fit	How mirrored content fills this destination when the shapes differ.
Reset	Returns the selection to an unscaled, unrotated, unflipped rectangle.

Corner handles scale the rectangle from the opposite corner, so the corner you are not dragging stays put.

The recording region gets two controls of its own, described under [the recording region](#).

Mirror

By default every destination shows its own slice of the canvas: whatever the arranger has placed under its rectangle. **Mirror** changes that. Set it to:

- **None**, the default. The destination shows what its rectangle covers.

- **Full canvas**, and the destination shows the whole arrangement, scaled to fit its frame. A Syphon or Spout share set this way hands the whole show to the receiving application rather than one screen's worth of it.
- **Another destination by name**, and this one repeats that one.

A mirroring destination stops reserving space on the canvas, because it is no longer showing a part of it. It cannot be dragged and its Position fields are locked, since it follows whatever it mirrors. Flip H and Flip V still apply.

Mirror Fit decides what happens when the mirrored content and the destination are different shapes:

Fit	What it does
Crop	Fills the frame and cuts off the overhang. The default.
Letterbox	Shows all of the mirrored content and draws black bars in the gap.
Contain	Shows all of it, with the surrounding canvas filling the gap instead of bars. The canvas grows to hold it.
Stretch	Distorts the mirrored content to fit the frame exactly.

Letterbox and Contain need somewhere to paint, so they are offered for a physical display and for the recording region only. Contain is not offered when mirroring the full canvas, because there is nothing outside the canvas to show.

If the destination you mirrored is turned off or disconnected, the row reads **Unavailable** and this destination goes back to showing its own slice until the other one returns. The setting is kept, not cleared.

A destination cannot mirror one that is itself mirroring something else.

Video Wall Controller Compensation

Set per destination, in the Transform panel. **Rows** and **Columns** describe the panel grid a hardware video wall controller splits this output into, and Visibox pre-compensates so the picture is not stretched.

Leave both at 1 (shown as Off) where no wall controller is involved. The full explanation, including how to work out the numbers, is under [Video walls](#).

Render Quality

These settings describe the machine rather than the show, so they stay put when you open a different Project.

Resolution Limit

Visibox renders once, into a single picture that every destination samples from. **Resolution Limit** caps how large that picture is: **Automatic**, **720p**, **1080p**, **4K**, or **8K**.

Automatic is the default and means no cap. Visibox renders enough pixels to serve the largest destination on the canvas at its full resolution.

Lower it when playback stutters on a machine driving several displays. The cap only ever reduces, never enlarges, and it keeps the shape of the picture exactly, so nothing is stretched. A display larger than the cap shows the smaller picture scaled up to fill it.

The limit reaches everything, not only screens. A recording, and an NDI, DeckLink, Syphon or Spout target, all take their picture from the same render.

Frame Rate Limit

Frame Rate Limit caps how many frames per second Visibox renders: **Automatic**, **24**, **30**, **60**, or **120** fps. Lower it when Visualizers or audio-reactive Effects stutter, since fewer frames a second gives each frame more time to finish. Video Clips play at their own frame rate and are unaffected.

Sharp rendering

Off by default. It matters only when you are driving more than one display and they do not all have the same pixel density, such as a laptop's high-resolution screen alongside a standard projector.

Left off, Visibox trims the render down to what the displays can actually show between them, which costs the high-resolution screen some detail. Turn it on and Visibox renders at the full resolution the high-resolution screen implies. The picture on that screen gains detail, and the machine does more work for it.

On a rig where every display has the same pixel density, this setting changes nothing.

Resolution Limit outranks it. A cap you set is applied after Sharp rendering has asked for more.

Letterbox Output window

Off by default. It changes how the Output window shows the canvas when the window is not the same shape as the canvas.

Off, the picture fills the window and the overhang is cropped. On, you see the whole canvas with bars where the shapes do not match.

This affects that one window and nothing else. What an NDI, DeckLink, Syphon or Spout target sends, and what a recording captures, are untouched by it. The same switch is on the Output window's right-click menu, which is the quicker route mid-show.

Ask before recording without a capture region

On by default. It controls the prompt that appears when you start a recording with no capture region set up. See [Recording the output](#).

Turn it back on here if you dismissed the prompt with "Don't ask again" and want it back.

Ask to exclude project folders from antivirus scanning

Windows only, and on by default. It controls the prompt that offers to exclude a Project folder from real-time antivirus scanning when Visibox notices a scanner reading the folder. See [Antivirus and file indexing](#).

NDI outputs

Add NDI Output creates a network video sender. It is the **NDI** tile in the Unused tray, and the new sender appears in that tray beside it, ready to drag onto the canvas. Turn on Settings > General > Enable Beta Features to see this.

Select it to set:

- **Name**, which is what receivers on the network see in their source lists.
- **Width** and **Height**, the resolution it sends at.
- **Remove NDI Output**, which asks for confirmation. If you change your mind, Edit > Undo Remove Output puts it back while Settings is still open.

DeckLink outputs (SDI)

Add DeckLink Output plays out over SDI through a Blackmagic DeckLink card. It is the **DeckLink** tile in the Unused tray, and it needs Blackmagic Desktop Video installed and a DeckLink device connected, and says so when no DeckLink device is found. With more than one card connected, a picker beside the tile chooses which one the new output uses. Turn on Settings > General > Enable Beta Features to see this.

Select a DeckLink rectangle to set its **Video Standard**, the display mode the card plays out. New outputs start on the card's first 1080p mode. The panel names the **Device** the output belongs to, and reports "Device not connected." when that card is not currently plugged in, so an output configured at home is still recognizable at the venue.

Remove DeckLink Output asks for confirmation. As with NDI, Edit > Undo Remove Output brings it back while Settings is still open.

Syphon, Spout, and the recording region

Three more destinations can appear on the canvas without being screens at all:

- **Syphon** (macOS) and **Spout** (Windows), once switched on in [Output Settings](#). Their rectangle decides how much of the canvas the shared source carries.
- The **Recording Capture Region**, which decides what a recording captures. It starts turned off, and while it is off recording captures the whole canvas, after asking you first. The region never enlarges the canvas, so placing it outside the other rectangles does not give them more picture to show.

Select the recording region and the Transform panel adds two controls the other rectangles do not have:

- **Recording Size** sets the resolution of the file: 720p, 1080p, 4K, 8K, or a width and height you type, which reads back as **Custom**. Changing it re-encodes the same area of the canvas at a different resolution. It does not move or resize the rectangle.
- **Match Canvas** snaps the rectangle to cover the whole canvas, keeping its shape. It is a one-time snap, not a link, so later changes to the canvas do not follow it.

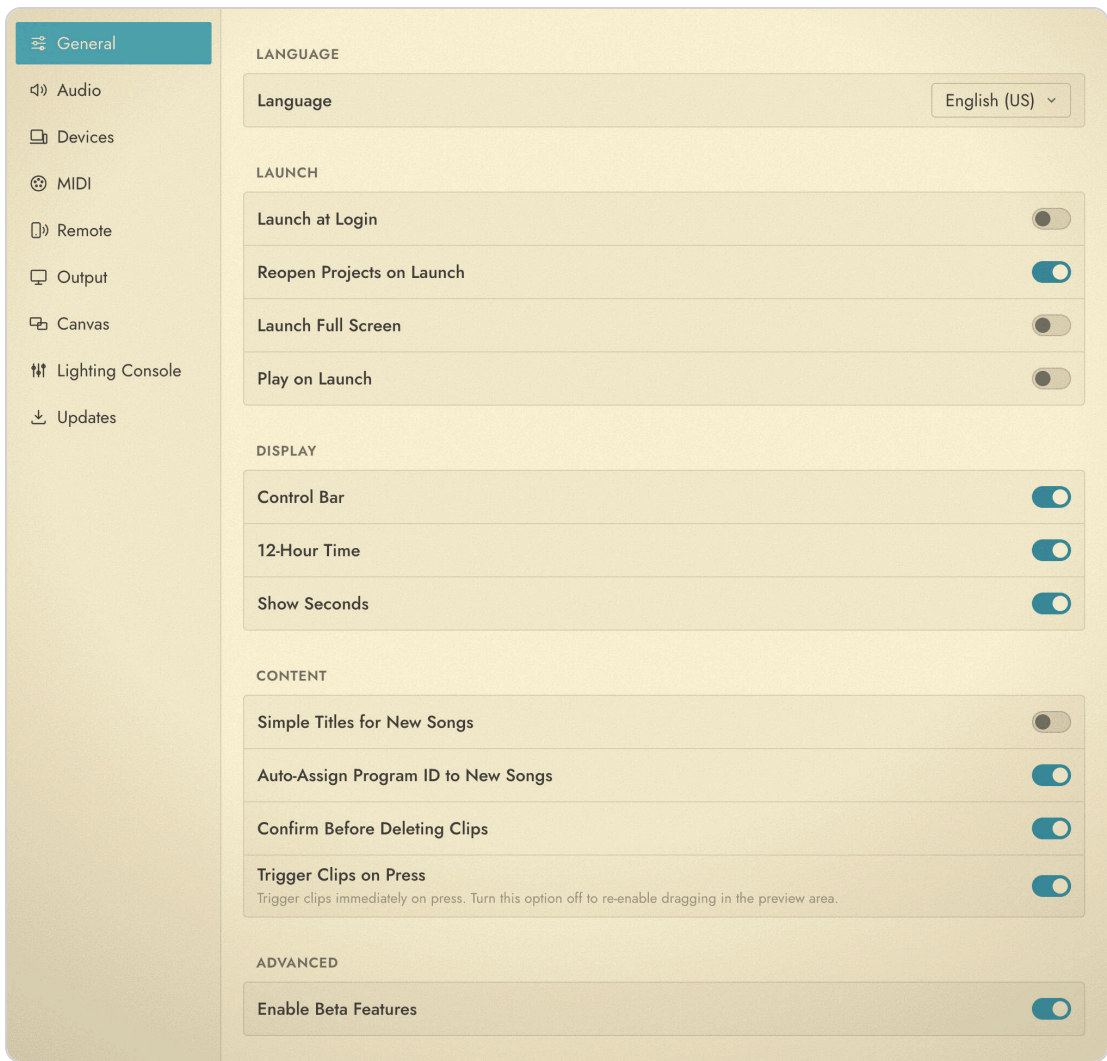
All three behave like any other rectangle: place them, size them, and turn them off again when you are done.

Identify Displays

Identify Displays flashes a large number on every connected screen for a few seconds. Use it when the arranger's rectangles and the actual screens in front of you have stopped matching up.

General Settings

The preferences that decide how Visibox behaves rather than what it plays.



General settings, grouped into Language, Launch, Display, Content and Advanced.

Language

Language sets the language of the Visibox interface. **System Default** follows your computer’s language. Everything repaints as soon as you pick, with no restart. This is the only place the language is set.

Launch

Setting	What it does
Launch at Login	Start Visibox when you log in to your computer.
Reopen Projects on Launch	Reopen the Projects that were open when you last quit. On by default.

Setting	What it does
Launch Full Screen	Start with the Output already full screen. Made for a machine that does nothing but run the show.
Play on Launch	Start playback as soon as the Project is open.

Launch at Login, Reopen Projects on Launch, Launch Full Screen and Play on Launch together make an unattended setup: power on, log in, and the show is running with nobody at the keyboard.

Display

Setting	What it does
Control Bar	Show or hide the Control Bar across the top of the Controller.
12-Hour Time	Show the clock as 12-hour instead of 24-hour.
Show Seconds	Add seconds to the clock.

Content

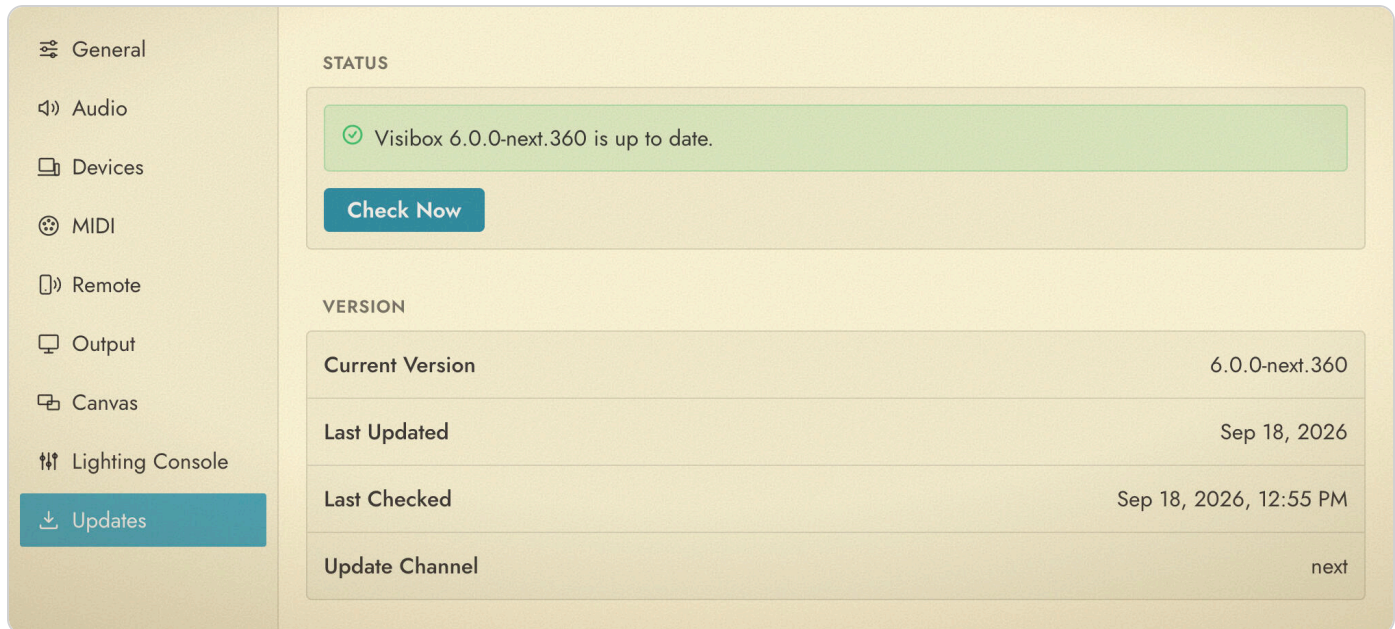
Setting	What it does
Simple Titles for New Songs	Name new Songs “New Song” instead of picking a random space-themed name.
Auto-Assign Program ID to New Songs	Give each new Song the next free Program Identifier, so MIDI program change can reach it without you setting one by hand. On by default.
Confirm Before Deleting Clips	Ask before a Clip is deleted. On by default.
Trigger Clips on Press	Fire a Clip the moment you press it rather than on release, in the Song list and in the Preview Pane alike. On by default.

Advanced

Enable Beta Features turns on the features that are still being finished. They work, but they change between releases, so leave this off on the machine you are playing a show with tonight.

Updates Settings

Which version of Visibox you are running, whether there is a newer one, and what changed in it.



Updates. The headline at the top answers the only question most people have.

Status

A single sentence at the top says where you stand, in one of a few forms:

- **Visibox x.y.z is up to date.**
- **Visibox x.y.z is available.** A newer build exists. Visibox asks before it downloads anything.
- **Visibox x.y.z is downloaded and ready to install.**
- **Visibox has not checked for updates yet.**
- **Could not check for updates.** The reason is printed underneath. On a Mac, one of those reasons is that Visibox is not in the Applications folder. Move it there and try again.

Check Now runs a check on demand. It reads “Checking...” while it works. Once an update has finished downloading, the button becomes **Relaunch to Update**, which quits Visibox, installs, and starts it again.

If your license covers updates up to a date rather than forever, a line here says which date. Past that date Visibox keeps running exactly as it is; only newer versions need a renewal.

When a check will not run

Visibox refuses to check for updates in situations where the answer could interrupt something, and it tells you which one applies:

- The Output, Controller or Lyrics window is full screen. Visibox does not check for updates during a show.
- An update dialog is already open, waiting for an answer.

- A check is already running.
- An update is already downloaded and only needs a relaunch.
- On Windows, the copy of Visibox you are running is not a real installation.

Relaunching is blocked by the same show rule for the same reason: it quits the app.

Version

The readout underneath the status.

Row	What it shows
Current Version	The build you are running. This is the number to quote in a support request.
Update Channel	The release stream this build takes updates from. A readout, not a choice.
Last Updated	When this version first launched, or Unknown .
Last Checked	When Visibox last asked the update server, or Never .
Latest Version	The version being offered, when one is.

What’s New

Release notes for the version on offer, so you can read what you are about to install before you install it. It reads “No release notes for this version.” when the server did not send any.

Visibox also checks on its own from time to time and offers the update as a dialog. **Check for Update...** in the menu bar is not the same as **Check Now** here: the menu item still runs while a window is full screen, and **Check Now** does not.

Troubleshooting

This page is arranged by what Visibox put on your screen. Find the message you saw, and the entry tells you what happened and which button to take.

Two things live elsewhere. If the problem is that playback is slow or soft rather than that something appeared, start at [Optimizing](#). If you want to reach a person, that is [Support & Bug Reports](#).

Opening a Project

“Invalid project file.”

Visibox could not read the Project’s data file.

If the Project has backups, the dialog offers **Open Latest Backup**. Take it. Visibox puts that backup in place of the damaged file and opens it, so you lose at most the five minutes since the last one. [Backups](#) covers how often Visibox writes them and what they hold.

If the Project has no backups yet, the dialog has nothing to offer and only lets you dismiss it.

The message also suggests copying the Project to a local drive. That is worth doing, because a cloud-synced folder is the most common way a Project file gets damaged: the sync client can be part way through writing the file at the moment Visibox reads it. Keep working Projects on a local disk and copy them elsewhere when you are done.

“Project file is locked.”

Another computer has this Project open. The dialog offers **Open Anyway** and **Cancel**, and Cancel is the default. Take the default. Two machines writing to one Project is how you lose work.

The lock is refreshed by the machine that holds it, and it goes stale after 30 minutes. So if the other computer crashed rather than quit, the Project frees itself half an hour after that machine last saved, and you can open it normally.

This is also why a Project on a shared drive is worth avoiding for anything you are actively editing. Visibox can tell you the file is in use, but it cannot stop two people opening it from two machines if they both click through.

“This project was saved by Visibox ...”

You are opening a Project that a newer version of Visibox has touched. The dialog names the version that saved it, offers **Open Anyway** and **Cancel**, and defaults to Cancel.

Before it asks, Visibox has already copied the file into the Project’s **Backups** folder, untouched. The copy is named `ORIGINAL`, followed in parentheses by the newer Visibox version that saved the Project. So if you do open the Project here and something is missing from it, the original is still on disk.

Open Anyway removes whatever the newer version added that this one cannot represent. If the Project matters, cancel, and open it on the machine with the newer Visibox instead.

NOTE

You will not always be asked. Visibox only raises the dialog when the newer file uses something this version would have to strip out. When the newer version changed nothing that matters here, the Project opens, and the untouched copy is put in **Backups** anyway.

“Media files are missing from your project’s Files directory.”

Visibox checks every media file when a Project opens. This dialog lists the ones it could not find, and offers three buttons.

- **Close Project** is the default and almost always the right answer. Quit, put the files back where they were, and open the Project again.
- **Remove Files from Project** drops the missing files **and deletes every Clip that used them**. Visibox asks a second time and tells you how many Clips that is. Read the number before you agree to it.
- **Continue** opens the Project with the files still missing. The Clips are all still there, and they will not play.

There is no relink step. Visibox will not search for a moved file, and it will not offer to point a Clip at one somewhere else. Putting the files back where they were is the only route.

A second, near-identical dialog appears when the files are present but cannot be read. That is usually a permissions problem or a drive that has gone away, and the same three buttons do the same three things.

Saving and disk space

“Your disk is full.”

Visibox ran out of room. Until you free some up, changes may not be written, and that includes both your Project and your sign-in.

Free space on the drive holding the Project, then carry on. Nothing needs restarting.

The message repeats while the disk stays full, but the gap between repeats doubles each time, up to half an hour. It cannot end up nagging you through a show.

“Not enough disk space.”

You are importing a file that will not fit. Visibox checks the space before it starts copying, so nothing is left half-imported.

A file that has to be converted on the way in is not checked up front. If the disk fills during that conversion, the import fails with a general error rather than this message. Free some space and try the import again.

“There was a problem saving the Project”

A save failed, and Visibox is telling you rather than carrying on quietly. The dialog carries the underlying error and offers **Close Project** and **Continue**.

The usual cause is a full disk, which has its own message above. If the disk has room, the next thing to suspect is where the Project lives: a network volume that dropped out, or a cloud folder that has locked the file.

Close Project is the safer answer. Continuing means every later change is also at risk of not being written.

Playback and windows

“GPU acceleration unavailable”

Clips are playing in software instead of on the graphics card, which is slower and can stutter. The dialog offers **Restart Visibox** and **Ignore**.

Restart Visibox. If the message comes back, restart the computer. If it comes back after that, your graphics drivers are the next thing to look at.

You see this once per session, and Visibox raises it early on purpose. The graphics system falls back to software after it has crashed a few times, and the fallback lasts for the rest of the session, so getting the warning before the doors open is the point of it. [It was fine yesterday and it is sluggish today](#) covers the same problem from the other end.

A window goes blank and comes back

If a Visibox window stops responding, Visibox waits a few seconds to confirm it is genuinely stuck rather than just busy, then reloads it. The whole thing takes about 20 seconds, and there is no dialog.

So a window that has gone white or frozen is worth 20 seconds of patience before you force quit anything.

Audio

“Audio Output Changed”

A notification, not a dialog, and it comes in three forms.

- Your chosen device is gone, and it names what the sound is playing through instead.
- Your chosen device is back, and Visibox has returned to it.
- There is no audio output at all.

Visibox moves the sound on its own rather than going silent, so the notification is telling you what already happened. If it named a device you do not want, pick the right one in [Settings > Audio](#).

Worth knowing before a show: a bus-powered interface that browns out for a moment will hand your audio to the built-in speakers, and this notification is the only thing that says so.

macOS system settings

Visibox checks two macOS settings that stop full screen working properly, and offers to fix each one.

“The ‘Displays have separate Spaces’ System Setting needs to be enabled.”

Without this, macOS will not put a full screen window on your second display the way Visibox needs.

The buttons are **Fix Automatically**, **Open System Settings** and **Remind Me Later**. Fix Automatically makes the change for you. Either way you have to log out and back in before it takes effect, and Visibox offers to log you out once the change is made.

Do this well before a show, not during load-in.

“Your System Settings are preventing the menu bar from hiding in full screen.”

The macOS menu bar is set to stay visible, so it sits on top of your full screen Output. The same three buttons appear, and this one takes effect immediately with no log out.

Still stuck

Help > Report a Bug sends us the details we need along with your message. If Visibox will not start at all, **Help > Copy Support Info** puts the same information on the clipboard so you can paste it into an email. See [Support & Bug Reports](#).

Support & Bug Reports

Visibox has a built-in form for reaching us. It opens in two modes: a general Support Form, and a Bug Report form with fields for the details we need to reproduce a problem.

Why the built-in forms help

When you submit through the built-in windows, Visibox automatically includes system information alongside your message: your operating system, Visibox version, hardware details, audio configuration, and other diagnostic data. This saves you from having to gather and report that information yourself, and means we can start investigating right away instead of asking follow-up questions.

If you email us directly, we will almost certainly need to ask for this information before we can help. The built-in forms skip that back-and-forth.

Support Form

Open from **Help > Visibox Support**.



Reach Out!

Got a question or need help? Send us a message. We'd love to hear from you! Our team speaks English but we've got Google Translate, so reach out in whatever language works best.

First Name

Sam

Last Name

Rivera

Email

sam@example.com

Issue

Can I run two projectors with different content from one Mac? We play a festival next month and want lyrics on one screen and visuals on the other.

Note: System information (OS, app version, etc.) will be included to help with troubleshooting.

Submit

The Support form. Ask a question or let us know how we can help.

Use the Support form for questions, feature requests, or anything else you want to tell us. Fill in your name, email, and a description of your question or issue, then click **Submit**. Our team speaks English but we have translation tools, so feel free to write in whatever language works best.

Bug Report Form

Open from **Help > Report a Bug**.

The image shows a 'Report a Bug' form with a yellow background. At the top center is a logo of a stylized eye with a rainbow-colored iris. Below the logo is the title 'Report a Bug' in bold black text. Underneath the title is a short instruction: 'Found a bug? Help us squash it! The more detail you provide, the faster we can fix it.' The form contains several input fields: 'First Name' with the value 'Sam', 'Last Name' with the value 'Rivera', and 'Email' with the value 'sam@example.com'. There are four larger text areas: 'Bug Description' containing 'The Output window goes black after I switch songs.', 'Steps to Reproduce' containing a numbered list of three steps, 'Expected Behavior' containing 'The next song starts playing on the Output.', and 'Actual Behavior' containing 'The Output stays black until I click the song again.' At the bottom left, a note states: 'Note: System information (OS, app version, etc.) will be included to help with troubleshooting.' At the bottom right is a blue 'Submit' button.



Report a Bug

Found a bug? Help us squash it! The more detail you provide, the faster we can fix it.

First Name

Last Name

Email

Bug Description

Steps to Reproduce

Expected Behavior

Actual Behavior

Note: System information (OS, app version, etc.) will be included to help with troubleshooting.

Submit

The Bug Report form. Structured fields help us reproduce and fix the issue.

The Bug Report form captures the details we need to reproduce and fix issues:

- **Bug Description:** a brief summary of what went wrong.
- **Steps to Reproduce:** what you did that triggered the bug. The more specific, the better.
- **Expected Behavior:** what you thought should happen.
- **Actual Behavior:** what actually happened instead.

TIP

The most helpful bug reports include specific steps to reproduce. “It crashed” is hard to investigate, but “I added a Visualizer Clip, set it to System Audio, then switched to fullscreen and it crashed” gives us something to work with.

Antivirus and file indexing

The first time you open a Project, Visibox checks whether something on your computer is going to read every file it writes. Video imports and automatic saves are large and frequent, and a scanner or indexer working through them competes with playback on the machine running the show.

If real-time scanning is switched on, Visibox offers to exclude the Project folder:

- **Fix Automatically** adds the exclusion for you. Windows asks you to approve it. This appears only when Windows Defender is the active scanner, because Visibox cannot change another product's settings.
- **Open Windows Security** takes you to the settings page so you can add the exclusion yourself. Go to Virus & threat protection, select Manage settings, then Add or remove exclusions, and add the Project folder.
- **Remind Me Later** dismisses the prompt. Visibox asks again the next time you open that Project.

Tick **Don't ask again** to stop the prompt for every Project. Only the folder you approve is excluded. The rest of your computer stays protected.

If a product other than Windows Defender is doing the scanning, Visibox names it and points you at Windows Security. Adding the exclusion inside that product is up to you.

Visibox handles this for you and does not ask. When a Project opens, it marks the Project's media and backup folders so Spotlight skips them. Your Project itself stays searchable, so a Project still turns up in a Spotlight search by name.

Copy Support Info

If you need to contact us outside of Visibox (for example, if Visibox will not launch), you can copy your system information to the clipboard from **Help > Copy Support Info** and paste it into an email.

Your Account

Everything about who you are signed in as, what you are entitled to, and which machines you have activated lives in one place: the Registration window.

Opening the Registration window

The Registration submenu is under **Visibox** on Mac and under **File** on Windows. What it contains depends on where you are:

- **Activate...** when Visibox does not know who you are yet.
- **Start Trial...** when you have not used your trial.
- **Enter Login Code...** when you have asked for a code and not entered it.
- **Trial Info...** during a trial.
- **Registration Info...** once you are signed in.
- **Purchase...** goes straight to the website.

All but the last open the same window.



Subscription Active

Hello,

Subscription Status: active

Subscription Type: annual

Subscription ID:

Last refresh: 9/18/26, 2:31 PM · [Refresh Now](#)

[Manage subscription](#)

[Deactivate Visibox on this device](#)

The Registration window. Your status is at the top, your plan details and account links at the bottom.

Signing in

Visibox does not use passwords. Enter your email address and it sends you a one-time code.



Activate

To start your free 14-day trial or activate an existing subscription, enter your email address:

SUBMIT

[I already have my login code...](#)

[I have a promo code...](#)

Purchase a Pro Plan

Unlock the full version with **no watermark** and **no time limit!**

GET A PRO PLAN

Signed out, the Registration window asks for your email address.

1. Open the Registration window and type your email address in the field.
2. Press **Submit**. Visibox sends a code to that address.
3. Type the code into **Enter One-time Code** and press **Submit**.

The code arrives from `no-reply@spaceage.tv`. If it does not turn up, check your spam folder. **Where is my login code?** in the window explains the same. **Wrong email address?** takes you back a step, and **I need a new code...** sends another.

If you have a promo code, click **I have a promo code...** before you submit and enter it alongside your email.

The free trial

New accounts get a **free 14-day trial** with everything switched on. The window's headline reads "Trial Version" and tells you how many days are left. The splash screen shows the same count each time you launch.

When the trial runs out the headline changes to "Trial Expired" and Visibox drops back to the free plan.

Free plan and Pro plan

Visibox keeps working after the trial. The **Free Plan** covers the basics with limits on how much a Project can hold and which features are available.

A **Pro plan** removes those limits and turns on the rest of the app. **GET A PRO PLAN** in the Registration window opens spaceage.tv/subscribe with your account already attached.

If you have bought a plan and Visibox has not caught up, click **I already have a pro plan...** to re-check with the server.

Managing your subscription

Once you are signed in, the lower half of the Registration window shows what you have:

- **Subscription Status** and **Subscription Type**
- **Subscription ID** (selectable, so you can copy it into a support message)
- **Last refresh**, with a **Refresh Now** link beside it

Manage subscription opens your account on spaceage.tv already signed in. Everything that changes money or machines lives there: your plan, your billing details, and the machines you have activated. Visibox hands your session across, so there is no second sign-in.

Payment past due

When a payment fails, the window turns to a warning and the headline reads **Subscription Payment Past Due**. A **Fix on website** link takes you straight to your billing details on spaceage.tv.

Subscription paused

A paused subscription reads **Subscription Paused** with the same **Fix on website** link, which lands on your plan so you can start it again.

Fix either one and click **Refresh Now** in Visibox, or reopen the Registration window, to pick the change up immediately.

Working offline

Visibox checks in with our servers from time to time to confirm your plan. It does not need to reach them to run.

When the machine has no internet, Visibox keeps working and quietly retries in the background. A venue with no usable network, a rehearsal room in a basement, a rig that travels: none of these stop a show.

If it goes long enough without reaching us, Visibox asks you to sign in again. **It will not interrupt a show to ask**, and a Project that is already loaded and playing carries on playing. There is no watermark and nothing shuts down mid-song.

Get the machine online for a moment, click **Refresh Now** in the Registration window, and everything comes back.



TIP

If you know you are going somewhere with no internet, open Visibox on a network first and let it check in. There is nothing else to arrange.

Devices

A plan covers a set number of machines. Activating Visibox on a computer uses one of them.

Deactivate Visibox on this device at the bottom of the Registration window frees the slot up. Use it before you sell or wipe a machine, and before you activate a replacement.

If you run out of slots, Visibox says so and offers **Manage devices**, which opens your device list on spaceage.tv. Deactivate one you are no longer using and try again.

Perpetual licenses

A perpetual license does not expire. The window reads **Perpetual License Active** and adds:

- **Purchased**, the date you bought it
- **Updates through**, the date your free updates run to

Visibox warns you in the window when that update window is within thirty days of closing, and **Renew upgrades** takes you to the website to extend it. Once the window closes, Visibox keeps running and keeps the version you have. Newer versions need a renewal.

Visibox Resources

Everything you need to get the most out of Visibox: community, tutorials, content, and support. Most of these are also accessible from the Help menu inside Visibox.

Community

Discord server

The [Visibox Discord server](#) is the best place to connect with other Visibox users and the Spaceage team. Ask questions, share your setups, show off your live visuals, and get tips from other performers. We are active on Discord and it is often the fastest way to get help.

Reddit community

The [Visibox subreddit](#) is a place to share your work, ask questions, and discuss all things Visibox with the community.

YouTube channel

The [Spaceage YouTube channel](#) has tutorials, feature walkthroughs, and performance videos. If you are a visual learner, this is a great place to start.

Learning

This manual

You are reading it! The Visibox Manual covers every feature in detail. Use the search bar at the top to find what you need quickly. Each version of Visibox has its own version of the manual, so the content always matches the version you are running.

Demo Project

Visibox includes a downloadable demo Project so you can explore the interface with real content without having to set up your own media first. Access it from **Help > Download/Open Demo Project**.

How-to articles

The [How-To articles](#) on the Spaceage website cover specific topics and workflows, from setting up your first show to advanced multichannel audio routing.

Frequently asked questions

The [Visibox FAQ](#) answers common questions about compatibility, licensing, hardware, and features.

Content

Video packs

Spaceage offers free [video loops and content packs](#) designed for live performance. These are a great way to build up your visual library quickly, and they work well for exploring Visibox features before you have your own content ready.

Where to find Clips and loops

Looking for more visual content? The article [Where to Find Video Clips and Loops for Visibox](#) lists content providers and sources for performance-ready video loops, including free and paid options.

Support

Built-in support and bug reports

Visibox has a built-in [Support Form](#), opened from **Help > Visibox Support** or **Help > Report a Bug**. It sends your system information along with the message, so we can start on the problem instead of asking for details. This is the best way to reach us.

Spaceage website

Visit spaceage.tv for the latest news, release announcements, and subscription management.

AI Assistants

You can control Visibox using natural language with an AI assistant like **Claude**, **Cursor**, **Codex**, **Windsurf**, **Gemini CLI**, **VS Code**, **Zed**, or **Cline**. Once connected, you talk to your show:

- “Add a Ken Burns effect to the first clip in Song 3.”
- “Create a new Song called Encore and add a Visualizer Clip.”
- “Set the volume to 50% and go fullscreen.”
- “Show me the songs in this project.”
- “Duplicate ‘Verse’ and rename the copy ‘Bridge’.”
- “Write a timeline for this song that stops everything at 2:30.”

Connecting is a one-time setup. After that you describe what you want, and you never have to think about the technology underneath.

How to Connect

The simplest way is to let Visibox auto-install itself into your AI tool of choice.

1. In Visibox, choose **Connect to AI Tools...** (on Mac, from the Visibox menu; on Windows, from the File menu).
2. Visibox scans for installed AI tools on your machine and offers to register itself with each one.
3. Restart your AI tool. Visibox is now available.

Once connected, ask the AI assistant about your Project. It discovers Visibox automatically and starts responding.

Supported Tools

Visibox can auto-install into any of these:

- Claude Desktop
- Claude Code
- Cursor
- Windsurf
- Codex
- Gemini CLI
- VS Code
- Zed
- Cline

For any other tool that supports the [Model Context Protocol](#), open [Settings > Remote](#) and click **Copy MCP URL** to get a URL you can paste into the tool's config.

What AI Assistants Can Do

An AI assistant connected to Visibox can read and change almost anything about your show:

Area	Examples
Playback	Play, stop, pause, resume, trigger a Clip, jump to a Song, seek to a position, set volume, go fullscreen.
Project state	Show the list of Songs and Clips, tell you what's currently playing, describe any Song or Clip.
Editing	Add, duplicate, move, rename, or delete Songs and Clips. Attach audio. Change fade, speed, or filters.
Effects	Apply an Effect to a Clip, remove one, or write a brand-new Effect in CSS from your description.
Visualizers	Add a Visualizer Clip, swap presets, or create a new ISF or MilkDrop preset from a prompt.
Song Timelines	Read a Song's timeline, write a new one, rewrite the one that's there, or clear it.

Because it's all natural language, the AI will fill in small details for you. You don't have to remember exact names or IDs. It's also fine to ask conversational questions like "which Song has the most Clips?" or "is there anything in this project that looks broken?"

Creating Effects and Visualizers with AI

Effects in Visibox are written in CSS, and Visualizers are written in GLSL/ISF (shader code) or MilkDrop. An AI assistant that understands those languages can:

- **Write a new Effect from a description.** Ask for "a warm VHS-like effect with subtle chromatic shift on the beat" and you get a real, working CSS Effect added to your library.
- **Modify an existing Effect.** "Make the Bass Pulse effect more aggressive on the peaks and shorter in duration."
- **Write a new Visualizer.** Describe the aesthetic and let the AI generate ISF shader code that you can drop into a Song.

When an AI edits an Effect or Visualizer, the changes show up **in real time** in the Effects Editor or Visualizers Editor if you have them open. You can watch code and the live preview update as the AI iterates, which makes creative back-and-forth feel natural.

Writing Song Timelines with AI

A [Song Timeline](#) is plain text, one cue per line, so it is a natural fit for an assistant. Ask for what you want in the show and let it write the cues:

- “Give Song 4 a timeline that plays clip 1 at the top, swaps to clip 3 at 1:45, and stops everything on the last chord at 4:12.”
- “Read me the timeline on the current Song and tell me what it does.”
- “Move every cue in this timeline two seconds later.”
- “Clear the timeline on Song 2.”

The assistant reads the existing timeline before it rewrites it, so you can iterate a cue at a time instead of dictating the whole sheet. Anything it writes shows up in the [Timeline Editor](#) if you have that window open, and **Edit > Undo** rolls it back.

Tips for Better Results

- **Open the Project first.** The AI works with the active Project, so make sure the one you want to control is open and frontmost.
- **Start simple.** Ask “what’s in this project?” to confirm the connection is working before issuing commands.
- **Be specific about names.** “Play the first Clip in Song 3” is clearer than “play the first clip.” Song and Clip titles help too.
- **Undo works.** If an AI-made edit isn’t what you wanted, use **Edit > Undo** ( **Z**) the way you would for any other edit.
- **You can refuse.** The AI has no secret powers. It uses the same operations you do through the menus. Most AI tools ask before acting, and **Edit > Undo** covers whatever slips through.

Local vs. Remote

By default, AI assistants running on the same computer as Visibox connect automatically, with no configuration. To connect an AI tool running on a different machine (a tablet, another laptop, a remote workstation), pair it first in [Settings > Remote](#).

Under the Hood

Visibox implements the [Model Context Protocol](#) (MCP), an open standard that lets AI assistants talk to external applications. The AI tool loads Visibox’s MCP server at startup and the two communicate over a local connection. What the AI tool then sends to its own AI service, and where that service runs, is up to that tool.

To integrate with Visibox at a lower level, see the [Visibox API](#) reference.

Command Line Interface

Visibox ships with a `visibox` command-line tool for controlling a running Visibox from your shell. It is a lighter alternative to the [AI Assistants](#) integration for quick one-offs, shell aliases, scripting, and hotkey tools.

The CLI is a thin wrapper over the same [HTTP REST API](#) the Stream Deck plugin and the MCP server use, so anything the CLI can do, any other client can too.

Installation

Install the CLI from inside Visibox:

1. Choose **Install CLI Tool...** (on Mac, from the Visibox menu; on Windows, from the File menu).
2. Visibox installs the command and shows you where it landed. Open a new terminal and run `visibox health` (Mac, or Windows from Command Prompt or PowerShell) or `visibox.cmd health` (Windows from Git Bash or WSL) to confirm it is working.

To remove it later, choose **Install CLI Tool...** again. Visibox detects that it is already installed and offers an **Uninstall** option.

Where it lands

Platform	Install path	Invoke as
macOS	<code>/usr/local/bin/visibox</code>	<code>visibox</code>
Windows	<code>%LOCALAPPDATA%\Visibox\bin\visibox.cmd</code>	<code>visibox</code> or <code>visibox.cmd</code>

On macOS, `/usr/local/bin` is already on your `$PATH`, so `visibox` works in any shell. You may be asked for your password, since writing there needs administrator rights.

On Windows, Visibox adds `%LOCALAPPDATA%\Visibox\bin` to your user `PATH`. In **Command Prompt** and **PowerShell**, type `visibox` and Windows resolves it to `visibox.cmd` automatically. In **Git Bash**, **WSL**, or another Unix-style shell on Windows, type `visibox.cmd` explicitly, because those shells do not do that resolution. To shorten it in bash, add `alias visibox=visibox.cmd` to your `~/.bashrc`.

The installed command is a small wrapper that runs the CLI out of the Visibox app itself, so there is no separate runtime to install.

 NOTE

You may need to open a new terminal window, or reload your shell profile, after installing so the command is on your `PATH`.

Quick Start

```
# Is Visibox running?
visibox health

# What's playing?
visibox status

# Control playback
visibox play
visibox stop
visibox next
visibox prev
```

Connection and Authentication

By default the CLI connects to `http://127.0.0.1:17736` on your local machine, and localhost connections skip authentication. There is nothing to configure for scripting your own Visibox.

To reach a Visibox on another machine you need a pairing token. See [Remote Pairing](#) for how to pair, then pass the token with a flag or an environment variable:

Flag / Variable	Purpose
<code>--host <host></code> / <code>VISIBOX_HOST</code>	Host to connect to.
<code>--port <port></code> / <code>VISIBOX_PORT</code>	Port. Default <code>17736</code> .
<code>--token <token></code> / <code>VISIBOX_TOKEN</code>	Pairing token for remote connections.

The CLI also reads a token from `~/.visibox/mcp-token` if it is there, the same file the MCP server uses, so if MCP already works then the CLI does too.

 NOTE

The default port changed in Visibox 6.0. Visibox 5.x used `51736`. Update any script that names the port.

Counting from One

Song and Clip indices are **one-based by default**, matching the numbers the Controller shows. `visibox play-clip 1` plays the first Clip.

This is the opposite of the [API](#), which is zero-based. The CLI takes the friendlier default because a person is typing it.

To switch, use `--zero-based` on a single command, set `VISIBOX_INDEX_BASE=0`, or make it permanent:

```
visibox config set index-base 0
```

`--one-based` and `--zero-based` cannot both be given at once.

Playback

Command	What it does
<code>visibox play</code>	Start playback.
<code>visibox stop</code>	Stop playback.
<code>visibox pause</code>	Pause.
<code>visibox resume</code>	Resume.
<code>visibox next</code>	Next Clip.
<code>visibox prev</code>	Previous Clip.
<code>visibox next-song</code>	Next Song.
<code>visibox prev-song</code>	Previous Song.
<code>visibox restart</code>	Restart the current Clip.
<code>visibox stop-all</code>	Stop all playing Clips.
<code>visibox play-toggle</code>	Toggle play and stop.
<code>visibox pause-toggle</code>	Toggle pause and resume.
<code>visibox play-clip <index></code>	Play the Clip at that position.
<code>visibox release-clip <index></code>	Release the Clip at that position.
<code>visibox song <index id></code>	Switch to a Song by position or ID.
<code>visibox trigger <clipId></code>	Trigger a Clip, respecting retrigger behavior.
<code>visibox release <clipId></code>	Release a Clip by ID.
<code>visibox seek <clipId> <percent></code>	Seek within a Clip, 0 to 100.

`play-clip` and `release-clip` both take `-s, --song <songId>` to work on a Song other than the active one.

`song` works out for itself whether you gave it a position or an ID. Force the reading with `--index <n>` or `--id <id>` if a Song ID of yours happens to be all digits.

Real-Time Control

Command	What it does
<code>visibox controls</code>	List the Project's Controls, their letters and their names.
<code>visibox control <letter> <value></code>	Set a Control by its letter, 0 to 1. Percentages work: 50%.
<code>visibox param <targetId> <value></code>	Set a bound parameter directly by its Target ID, skipping Controls.

See [Real-Time Control](#) for what Controls and Targets are.

Releasing a Clip

`release` and `release-clip` are the note-off half of a held button, new in 6.o. They stop a Clip that is in [Gate launch mode](#) and do nothing to a Clip in any other launch mode.

That makes them safe to fire unconditionally, which is what a hotkey tool wants:

```
# on key down
visibox play-clip 3
# on key up
visibox release-clip 3
```

System

Command	What it does
<code>visibox fullscreen</code>	Toggle the Output window full screen.
<code>visibox mute</code>	Toggle output mute.
<code>visibox volume <0-100></code>	Set the master output level.
<code>visibox panic</code>	Emergency reload. Restart all Visibox windows.
<code>visibox undo</code>	Undo the last project edit.
<code>visibox redo</code>	Redo the last undone project edit.

Query

Command	What it returns
<code>visibox health</code>	Whether Visibox is running and reachable.
<code>visibox status</code>	The active project and playback state.

Command	What it returns
<code>visibox app-state</code>	Full screen, mute, and volume.
<code>visibox projects</code>	All open projects.
<code>visibox project [id]</code>	Detail for a project. Defaults to the active one.
<code>visibox songs</code>	Songs in the active Project.
<code>visibox clips <songIndexOrId></code>	Clips in a Song.
<code>visibox presets</code>	Available Visualizer and Effect presets.
<code>visibox cameras</code>	Connected cameras and video inputs.
<code>visibox version</code>	CLI, app, and API versions from the running Visibox.

`presets` takes `-k, --kind <effect\|visualizer>` and `-s, --source <app\|user\|plugin>` to narrow the list.

`visibox --version` is different: it prints the CLI's own version without contacting Visibox.

Editing

A single `edit` command dispatches any edit the API supports:

```
visibox edit song:update-title --param songId=abc123 --param title="New Title"
visibox edit clip:delete --param clipId=xyz789
```

Run `visibox edit list` to print every available edit name, grouped by kind. Running `visibox edit` with no name does the same thing.

To see what one edit takes:

```
visibox edit song:set-timeline --describe
```

For nested parameters, pass the whole object at once instead of repeating `--param`:

```
visibox edit clip:update-options --params-json '{"clipId":"xyz789","options":
{"speed":1.5}}'
```

A `--param` value that parses as JSON is coerced, so `--param volume=0.5` sends a number and `--param mute=true` sends a boolean.

Edit parameters are always zero-based, whatever your index-base setting is.

JSON Output

Append `--json` to any command for machine-readable output instead of formatted text. This is the right mode for piping into `jq`, `grep`, or another script:

```
visibox status --json | jq '.project.name'
visibox songs --json | jq -r '.[].title'
```

With `--json`, errors go to standard error as JSON too, so a script can tell an application error from a connection failure without parsing prose.

Targeting a Specific Project

Most commands default to the active Project. Use `--project <id>` to target another:

```
visibox songs --project abc123
visibox play --project abc123
```

Global Options

Option	Description
<code>--json</code>	Output raw JSON instead of formatted text.
<code>--host <host></code>	Visibox host. Default <code>127.0.0.1</code> .
<code>--port <port></code>	Visibox port. Default <code>17736</code> .
<code>--token <token></code>	Auth token for remote connections.
<code>--project <id></code>	Target Project ID. Defaults to the active Project.
<code>--timeout <ms></code>	Request timeout in milliseconds. Default <code>5000</code> .
<code>--one-based</code>	Treat Song and Clip indices as one-based. This is the default.
<code>--zero-based</code>	Treat Song and Clip indices as zero-based.
<code>--version</code>	Print the CLI version and exit.

Configuration

`visibox config` reads and writes preferences in `~/.visibox/config.json`:

```
visibox config get
visibox config get index-base
visibox config set index-base 0
```

`index-base` is the only key, and it takes `0` or `1`.

Exit Codes

Code	Meaning
<code>0</code>	Success.
<code>1</code>	Application error. Visibox returned an error, or refused the action.
<code>2</code>	Connection error. Visibox is not running or is unreachable.

Exit codes make the CLI friendly to shell scripts: check `$?` or chain with `&&` and `\| \|` like any other Unix tool.

NOTE

As of 6.0, an action Visibox refuses exits `1` rather than `0`. A script that checked only for a connection failure now catches a refused action too.

Examples

The examples below are written for bash (macOS, Linux, WSL, Git Bash). On Windows, run them from Git Bash or WSL, or translate the loops and `jq` pipelines to PowerShell. In Git Bash, substitute `visibox.cmd` for `visibox`.

```

# Fire off a panic reload from a Stream Deck button, keyboard shortcut tool, or
hotkey app
visibox panic

# Rotate through songs on a timer (for example, in a soundcheck script)
for i in 1 2 3 4 5; do
    visibox song $i
    sleep 30
done

# Dump a set list for a show document
visibox songs --json | jq -r '.[.] | "\(.index). \(.title)'"

# Trigger the first Clip of every Song in sequence (hands-off preview)
visibox songs --json | jq -r '.[].id' | while read songId; do
    visibox song "$songId"
    visibox play-clip 1
    sleep 15
done

# Bail out of a script if Visibox isn't running
visibox health || { echo "Visibox is not running"; exit 1; }

```

TIP

The CLI, the [Stream Deck plugin](#), and [AI Assistants](#) are three ways to reach the same API. Reach for the CLI when you want scripting or shell-speed control, the Stream Deck for physical buttons, and an AI assistant when you want to drive Visibox with natural language.

Visibox API

Visibox exposes a local network API for remote control, automation, and custom integrations. This page is the technical reference. If you want to use Visibox through an AI assistant or a shell, see [AI Assistants](#) or [Command Line Interface](#) instead. If you want to drive it from a control surface, see [OSC](#).

Overview

The API runs in a separate process for stability. If your integration misbehaves, it does not affect Visibox itself. Three services are available:

Service	Port	Protocol	Description
WebSocket	17734	<code>ws://</code>	Real-time two-way communication. State updates are pushed to connected clients.
HTTP REST	17736	<code>http://</code>	Request and response, for querying state and sending commands.
OSC	17735	UDP	Open Sound Control. See OSC .

All three bind to every network interface on the machine, so any device on the same local network can connect.

 NOTE

These port numbers changed in Visibox 6.0: WebSocket moved from 51734 to 17734, OSC from 51735 to 17735, and HTTP from 51736 to 17736. If you have an integration built against the older numbers, update them.

Service Discovery

Visibox advertises itself on the local network using Bonjour (mDNS/DNS-SD):

Service Type	Advertised as	Port
<code>_visibox._tcp</code>	Visibox - <hostname>	17734
<code>_http._tcp</code>	Visibox HTTP - <hostname>	17736

The TXT records include the API version, hostname, platform, and the port numbers, including `oscPort`. OSC has no Bonjour service of its own.

Authentication

Localhost

Visibox authenticates connections from `127.0.0.1` or `::1` automatically. No pairing and no token.

Integrations running on the same machine, such as the Stream Deck plugin, the CLI, and the MCP server, connect with no setup.

Remote devices

Devices connecting over the network must pair first. See [Remote Pairing](#) for how to pair a device and how long the token lasts.

Once paired, the device includes its token in every request, either as an `Authorization: Bearer <token>` HTTP header or in the initial WebSocket `auth` message.

HTTP REST API

Base URL: `http://<visibox-ip>:17736`

Health and info

```
GET /health
GET /api/v1/info
```

`/health` needs no authentication and is the quickest way to check the API is up. `/api/v1/info` returns the API version, the app version, and a map of every endpoint.

Pairing

```
POST /api/v1/pair
POST /api/v1/refresh
```

`POST /api/v1/pair` exchanges a 6-digit code for a token:

```
{
  "otp": "123456"
}
```

Returns:

```
{
  "token": "eyJhbGciOiI..."
}
```

Projects

```
GET /api/v1/projects
GET /api/v1/projects/:projectId
GET /api/v1/projects/:projectId/summary
GET /api/v1/projects/:projectId/playstate
GET /api/v1/active
GET /api/v1/app-state
```

`/projects` lists every open project. `/projects/:projectId` returns everything: Songs, Clips, files, Effects, Visualizers, and display options. `/summary` is the light version of that, for a client that only needs names and counts.

`/playstate` returns what is playing, with positions, durations, and directions. `/active` bundles the active project and its play state into one request. `/app-state` returns full screen, mute, and volume.

Drill-down endpoints

For part of a Project without fetching the whole thing:

```
GET /api/v1/projects/:projectId/songs
GET /api/v1/projects/:projectId/songs/:songId
GET /api/v1/projects/:projectId/clips?ids=<id>,<id>
GET /api/v1/projects/:projectId/effects
GET /api/v1/projects/:projectId/visualizers
```

`/clips` takes a required `ids` query parameter and returns only those Clips.

Presets

```
GET /api/v1/presets
GET /api/v1/presets/:kind/:id
```

List the Effect and Visualizer presets, or fetch one. `:kind` is `effect` or `visualizer`.

Media

```
GET /thumbnails/:projectId/:fileId.jpg
```

A Clip's thumbnail. Optional query parameters:

Parameter	Default	Description
<code>size</code>	<code>original</code>	Square output size in pixels (16 to 512).
<code>over lay</code>	<code>none</code>	Overlay icon: <code>play</code> , <code>pause</code> , or <code>none</code> .

Parameter	Default	Description
<code>format</code>	jpeg	Output format: <code>jpeg</code> or <code>png</code> .
<code>quality</code>	85	JPEG quality, 1 to 100.

```
GET /waveforms/:projectId/:fileId.png
```

A waveform image for an audio file.

Cameras

```
GET /api/v1/cameras
GET /camera-thumbnails/:cameraId.jpg
```

List cameras with their device IDs and labels, or fetch a live thumbnail, updated every 5 seconds while the camera is active. Use `default` as the camera ID for the default camera.

Sending actions

```
POST /api/v1/action
```

The body names the action and its parameters:

```
{
  "name": "PLAY_CLIP_BY_INDEX_IN",
  "params": ["0", "1"],
  "base": 0
}
```

Field	Required	Description
<code>name</code>	Yes	An action name from the list below.
<code>params</code>	No	Array of values. Everything is coerced to a string.
<code>projectId</code>	No	Target project. Omit for the active one.
<code>base</code>	No	<code>0</code> or <code>1</code> . See Counting from zero or one .

Editing a Project

```
POST /api/v1/edit
GET  /api/v1/edit/schema
GET  /api/v1/edit/schema/:name
```

`POST /api/v1/edit` applies a Project mutation: add, move, or delete Songs, Clips, Effects, and Visualizers, set a Song's timeline, import media, undo, redo. The two schema endpoints list the available edit names and the parameters each one takes, which is the reliable way to discover them rather than hard-coding a list.

Uploading a file

```
POST /api/v1/files/upload
```

WebSocket API

Connect to `ws://<visibox-ip>:17734/v1`. The `/v1` path is required.

The WebSocket API is two-way. Connect once, authenticate, and state updates are pushed to you as things change, with no polling.

Connecting

As soon as the socket opens, and before you have sent anything, the server pushes a `connected` event carrying your client ID. It is a greeting, not an acknowledgement: every WebSocket connection authenticates, localhost included.

Send an `auth` message next:

```
{
  "type": "auth",
  "payload": {
    "token": "your-token"
  }
}
```

On success the server replies with a `response` message carrying the same `id` as your `auth` message and `data: { "authenticated": true }`. Wait for that before you send anything else. A remote device pairing for the first time sends `otp` and `deviceName` instead of `token`. A client on the same machine can send `auth` with neither: Visibox puts an approval prompt on screen, and the `response` that follows carries a `token` to reuse next time. See [Remote Pairing](#).

Receiving state updates

The server pushes `state-update` events:


```

{
  "type": "event",
  "event": "state-update",
  "payload": [
    {
      "updateType": "playstate",
      "projectId": "p1234567890",
      "data": {
        "event": "clip-play-state-changed",
        "clipID": "c001",
        "newState": "playing"
      }
    }
  ]
}

```

`event` sits beside `payload`, not inside it, and `payload` is the array of updates itself. There is no wrapper key around it.

Update types are `project`, `playstate`, `clip`, `song`, `volume`, `appState`, `presets`, and `full`. `appState` carries app-level changes (full screen, mute, output level) and its `projectId` is `null`, because it belongs to no one project. `presets` fires when an Effect or Visualizer preset is added, changed or removed, so a client holding a cached preset list knows to refresh it. Updates are coalesced, so a burst of changes arrives as one message rather than fifty.

Sending actions

```

{
  "id": "req-123",
  "type": "action",
  "action": {
    "name": "NEXT_SONG_IN",
    "params": []
  },
  "base": 1
}

```

The server responds with a `response` message carrying the same `id`, so you can match responses to requests. `base` sits beside `action`, not inside it.

Available Actions

Action names are uppercase and go in the `name` field over any transport.

Playback

Action	Parameters	Description
<code>PLAY_IN</code>	none	Start playback.
<code>STOP_IN</code>	none	Stop playback.
<code>PAUSE_IN</code>	<code>clipID</code> (optional)	Pause.
<code>RESUME_IN</code>	<code>clipID</code> (optional)	Resume from a pause.
<code>PLAY_TOGGLE_IN</code>	none	Toggle play and stop.
<code>PAUSE_TOGGLE_IN</code>	none	Toggle pause and resume.
<code>PLAY_CLIP_IN</code>	<code>clipID</code> (optional)	Play a Clip by ID.
<code>PLAY_CLIP_BY_INDEX_IN</code>	<code>index</code> , <code>songID</code> (optional)	Play a Clip by its position.
<code>TRIGGER_CLIP_IN</code>	<code>clipID</code> , <code>activateSong</code> (optional)	Trigger a Clip, respecting its retrigger behavior.
<code>RESTART_CLIP_IN</code>	<code>clipID</code> (optional)	Restart a Clip from the beginning.
<code>RELEASE_CLIP_IN</code>	<code>clipID</code>	Release a held Clip. See below.
<code>RELEASE_CLIP_BY_INDEX_IN</code>	<code>index</code> , <code>songID</code> (optional)	Release a held Clip by its position.
<code>STOP_CLIP_IN</code>	<code>clipID</code> , <code>maybePlayBackground</code> (optional)	Stop one Clip.
<code>STOP_ALL_CLIPS_IN</code>	<code>maybePlayBackground</code> (optional)	Stop every Clip.

`RELEASE_CLIP_IN` and `RELEASE_CLIP_BY_INDEX_IN` are new in 6.0. They are the note-off half of a held button, and they do nothing unless the Clip is in [Gate launch mode](#). That makes them safe to send unconditionally: build your surface to send trigger on press and release on release, and Clips in the other launch modes ignore the release.

Navigation

Action	Parameters	Description
<code>NEXT_CLIP_IN</code>	<code>anySong</code> (optional)	Next Clip. Pass <code>true</code> to cross into the next Song.
<code>PREV_CLIP_IN</code>	<code>anySong</code> (optional)	Previous Clip.
<code>NEXT_SONG_IN</code>	<code>noStop</code> (optional)	Next Song. Pass <code>true</code> to leave playback running.
<code>PREV_SONG_IN</code>	<code>noStop</code> (optional)	Previous Song.
<code>PLAY_NEXT_SONG_IN</code>	none	Move to the next Song and start playing it.

Action	Parameters	Description
SONG_SELECT_IN	index, stopPlaying (optional)	Select a Song by its position.
SONG_SELECT_BY_ID	songID, stopPlaying (optional)	Select a Song by ID.
SONG_SELECT_PGM_ID	pgmId, stopPlaying (optional)	Select a Song by its Program Identifier.
SONG_SELECT_PERCENT_IN	percent, stopPlaying (optional)	Select a Song by position through the list, 0.0 to 1.0.

stopPlaying defaults to true.

Seeking and direction

Action	Parameters	Description
SEEK_CLIP_IN	clipID, percent	Seek a Clip. percent is 0.0 (start) to 1.0 (end).
SEEK_SONG_IN	songID, percent	Seek Song audio, 0.0 to 1.0.
CLIP_DIRECTION	clipID, direction	Set direction: forward or backward.

WARNING

Both seek actions take a fraction from 0.0 to 1.0, not a number of seconds. Older documentation described them as seconds. The [CLI seek](#) command takes 0 to 100 and converts, so the two differ on purpose.

Real-Time Control

Action	Parameters	Description
SET_CONTROL	controlID, value	Drive a Control.
SET_PARAM	targetID, value	Write one Target directly, skipping Controls.

Both are new in 6.0. controlID is a Control's letter, A through Z, case-insensitive. targetID is of the form clip:abc123/opacity. Values are conventionally 0.0 to 1.0 and are not clamped here: shaping happens in the Control's own settings. See [Real-Time Control](#).

System

Action	Parameters	Description
OUTPUT_LEVEL_IN	level (0.0 to 1.0)	Set the master output level.
MUTE_TOGGLE	none	Toggle mute.
FULLSCREEN_TOGGLE	none	Toggle full screen.
RECORD_TOGGLE	mode (optional)	Toggle recording the output to a file.
PANIC	none	Reload every window.
OPEN_EDITOR	kind, presetID	Open the Effects or Visualizers editor on a preset. kind is effect or visualizer.

RECORD_TOGGLE is new in 6.0. When nothing is recording, mode chooses how it starts: now begins immediately, next-trigger arms it to begin on the next Clip trigger. When recording is already armed or running, the action disarms or stops it and mode is ignored.

Velocity

Actions that start a Clip carry an optional velocity field alongside params, from 0 to 127, matching a MIDI note-on. It applies to TRIGGER_CLIP_IN, PLAY_CLIP_IN, PLAY_CLIP_BY_INDEX_IN, and RESTART_CLIP_IN, and feeds the Clip's Velocity Sensitivity setting. Omit it and the trigger counts as full velocity.

Counting from zero or one

Clip and Song positions are zero-based by default: the first Clip is 0. The Controller numbers them from one.

Send base: 1 alongside an action and Visibox subtracts one from the first parameter of PLAY_CLIP_BY_INDEX_IN, RELEASE_CLIP_BY_INDEX_IN, and SONG_SELECT_IN, so your integration can use the numbers your user sees. Those three are the only actions it touches, and it does not apply to edit parameters, which are always zero-based.

Send the release with the same base you sent the trigger with. A gate button whose press and release disagree on the base holds one Clip down and lets go of its neighbor.

MCP Tools Reference

The [AI Assistants](#) integration is powered by an MCP server exposing tools to AI clients. It is served from the same HTTP port at /mcp, with discovery at GET /.well-known/mcp.json. Three methods are registered on that path: POST sends a request, GET opens a server-sent-events stream for messages coming back, and DELETE ends the session. An MCP client library handles all three for you. For end-user setup, see the AI Assistants page. This section covers the tools themselves.

Query tools

Tool	Description
<code>visibox_get_projects</code>	List all open Projects.
<code>visibox_get_project</code>	Full data for a Project: Songs, Clips, files, Effects, Visualizers.
<code>visibox_get_project_summary</code>	Names and counts only, for a cheap overview.
<code>visibox_get_active_project</code>	The active Project and its play state in one call.
<code>visibox_get_playstate</code>	Play state for a Project.
<code>visibox_get_app_state</code>	App-level state: full screen, mute, volume.
<code>visibox_get_songs</code>	List Songs in a Project.
<code>visibox_get_song</code>	One Song, including its timeline and lyrics.
<code>visibox_get_clips</code>	Several Clips by ID.
<code>visibox_get_clip</code>	One Clip.
<code>visibox_get_effects</code>	Effects in a Project.
<code>visibox_get_visualizers</code>	Visualizers in a Project.
<code>visibox_get_presets</code>	List Effect and Visualizer presets.
<code>visibox_get_preset</code>	One preset by ID and kind.
<code>visibox_get_preset_controls</code>	The parameters a preset exposes.

Edit tools

Tool	Description
<code>visibox_clip_edit</code>	Add, move, duplicate, delete Clips, or update Clip options including filters and effects.
<code>visibox_song_edit</code>	Add, move, duplicate, delete Songs, manage Song audio and titles, set Program Identifiers, and set a Song's timeline.
<code>visibox_project_edit</code>	Undo, redo, display options, Background Clips, zoom.
<code>visibox_effect_edit</code>	Add, update, remove, or copy Effects.
<code>visibox_visualizer_edit</code>	Add, update, remove, or copy Visualizers.
<code>visibox_preset_edit</code>	Add, update, or remove user Effects and Visualizers in the preset store.
<code>visibox_file_edit</code>	Import media files.
<code>visibox_describe_edit</code>	Return the parameter schema for one edit name.
<code>visibox_validate_preset</code>	Check a preset before writing it.

Action tools

Tool	Description
<code>visibox_playback</code>	Playback controls: play, stop, pause, resume, trigger Clips, navigate, seek, volume.
<code>visibox_system</code>	System controls: full screen, mute, recording, panic.
<code>visibox_open_editor</code>	Open the Effects or Visualizers editor on a given preset.
<code>visibox_control</code>	Real-Time Control: list the Controls, set one by letter, or set a parameter by Target ID.

The MCP tools cover most but not all of the action list. Anything missing can be driven over HTTP, WebSocket, or [OSC](#).

Live editor updates

When `visibox_effect_edit` or `visibox_visualizer_edit` modifies a preset, the matching Editor window updates in real time if it is open. Code changes, parameter changes, and the live preview all reflect the edit as it happens.

Stream Deck Plugin

Visibox has an official Elgato Stream Deck plugin that connects through this API. See the [Stream Deck](#) page for full details.

Security

The API is designed for local network use:

- All communication is unencrypted. Do not expose these ports to the public internet.
- Localhost connections bypass authentication by design, for integrations on the same machine.
- One-time pairing codes expire after 5 minutes.

Error Handling

HTTP status codes

Code	Meaning
400	Bad request. Invalid parameters.
401	Authentication required, or an invalid token.
403	Your plan does not include API access.
404	Resource not found.

Code	Meaning
500	Server error, or the action was refused.
503	Service unavailable. The API is not ready yet.

An action Visibox refuses comes back as a 500 whatever the reason, so read the `error` string rather than branching on the status code.

WebSocket error types

Type	Description
<code>INITIALIZATION_FAILED</code>	The server could not start.
<code>CONNECTION_FAILED</code>	The connection could not be established.
<code>AUTHENTICATION_FAILED</code>	Invalid token.
<code>ACTION_FAILED</code>	The action could not be executed.
<code>INVALID_MESSAGE</code>	Malformed message.
<code>RATE_LIMITED</code>	Too many requests.
<code>SERVER_ERROR</code>	Internal server error.

OSC

Visibox listens for **OSC** (Open Sound Control) messages over the network, so anything that speaks OSC can drive it: a phone running TouchOSC, a QLab cue stack, Resolume, a Max patch, a Lemur layout, or a lighting console with an OSC module. Visibox can also send its state back, so a fader on your tablet moves when someone changes the volume in the app.

If you are choosing between integrations: OSC is the right one for a purpose-built control surface and for cue systems that already talk OSC. For scripting from a shell, see [Command Line Interface](#). For an application you are writing yourself, see [Visibox API](#).

Connecting

Protocol	OSC 1.0 over UDP
Port	17735, unless you change it under OSC Port in Settings > Remote
Address	Your computer's IP address on the local network

Point your OSC app at your computer's IP address and port 17735. Everything on this page goes to that one address, and the examples below all assume the default port.

Discovery

Visibox advertises itself on the local network with Bonjour, under the service type `_visibox._tcp`. The OSC port is published as an `oscPort` field in that service's TXT record. There is no separate `_osc._udp` service, so an app that browses only for OSC services will not find Visibox. Point it at the IP address and port instead.

Pairing

Every OSC sender has to be a paired device first, and that includes an app on the same machine as Visibox.

OSC has no pairing step of its own. Pair the device over the regular API connection first, and pairing it authorizes its IP address for OSC as well. See [Remote Pairing](#). Once the IP is paired, OSC messages from it are accepted with no token in the message.

If OSC seems to be doing nothing at all, an unpaired sender is the first thing to check.

The Address Space

Playback

Address	Arguments	What it does
<code>/visibox/play</code>	none	Start playback.
<code>/visibox/stop</code>	none	Stop playback.

Address	Arguments	What it does
/visibox/pause	none	Pause.
/visibox/resume	none	Resume from a pause.
/visibox/play-toggle	none	Toggle between play and stop.
/visibox/pause-toggle	none	Toggle between pause and resume.

Clips

Address	Arguments	What it does
/visibox/clip/trigger	index (int), songID (string, optional)	Trigger a Clip by its position.
/visibox/clip/release	index (int), songID (string, optional)	Release a Clip. See below.
/visibox/clip/next	anySong (int, optional)	Next Clip. Pass 1 to cross into the next Song.
/visibox/clip/prev	anySong (int, optional)	Previous Clip.

/visibox/clip/release is the other half of a held button. It does nothing unless the Clip is in Gate launch mode, so a button that sends trigger on press and release on release is safe to point at any Clip. Gate is covered on the [Launch Modes](#) page.

Songs

Address	Arguments	What it does
/visibox/song/select	index (int), stopPlaying (int, optional)	Select a Song by its position.
/visibox/song/next	noStop (int, optional)	Next Song. Pass 1 to leave playback running.
/visibox/song/prev	noStop (int, optional)	Previous Song.

Volume

Address	Arguments	What it does
/visibox/volume	level (float, 0 to 1)	Set the master output level.

Real-Time Control

Address	Arguments	What it does
<code>/visibox/control/<letter></code>	<code>value</code> (float)	Drive a Control.
<code>/visibox/param/<targetId></code>	<code>value</code> (float)	Write one Target directly.

`<letter>` is a Control's letter: `/visibox/control/A`, `/visibox/control/b`, and so on. Case does not matter. Addressing by letter is the portable way to do it, because a layout you built for one project still lands on the right Control in another.

`<targetId>` is everything after `/visibox/param/`, slashes and colons included.

`/visibox/param/clip:abc123/opacity` addresses the Target `clip:abc123/opacity`. Use this when you want to skip Controls and write a parameter straight from your surface.

Both take a real number: an OSC float or int. A string is rejected even when it looks like a number, because an empty string would read as zero and black out a Control bound to opacity mid-show. Send a float.

Neither address is clamped at this end. A Control is conventionally driven 0 to 1, and the shaping happens in the Control's own settings. See [Real-Time Control](#).

System

Address	Arguments	What it does
<code>/visibox/panic</code>	none	Reload every Visibox window.
<code>/visibox/fullscreen</code>	none	Toggle full screen.

Counting from zero or one

Clip and Song positions count from zero by default, so the first Clip is `0`. The Controller shows them numbered from one. To make your surface match what you see on screen, send this once:

```
/visibox/config/index-base 1
```

The setting applies to `/visibox/clip/trigger`, `/visibox/clip/release`, and `/visibox/song/select`, and only for the sender that asked for it. Visibox remembers the opt-in by the sender's address and port, so an app that reconnects from a different port needs to send it again. It does not change the numbers Visibox sends back in state messages, which stay zero-based. It does not affect the Real-Time Control addresses, which do not use positions at all.

Each way into Visibox counts separately. OSC and the [API](#) are zero-based unless you ask otherwise; the [command line tool](#) is one-based unless you ask otherwise. Set the base explicitly on anything you build and you never have to remember which is which.

State Feedback

Ask Visibox to send its state back and a fader on your surface can follow what the app is doing.

Subscribing

```
/visibox/subscribe <replyHost:string> <replyPort:int>
```

You can shorten it. Send one integer and Visibox uses your IP address with that port. Send nothing at all and it replies to the address and port the message came from, which is what most surfaces want.

To stop, send `/visibox/unsubscribe`.

Visibox tracks up to 32 subscribed endpoints at once. A 33rd pushes out the oldest.

What comes back

Address	Arguments	Sent when
<code>/visibox/state/playback</code>	<code>status</code> (string)	Playback changes. <code>playing</code> , <code>paused</code> , or <code>stopped</code> .
<code>/visibox/state/song</code>	<code>songID</code> (string)	The active Song changes.
<code>/visibox/state/volume</code>	<code>level</code> (float)	The master output level changes.

Playback and volume are sent as soon as you subscribe, so your surface can catch up without waiting for something to move. These go out when a value actually changes, not on a timer.

Errors

A Real-Time Control message Visibox refuses is answered rather than ignored:

```
/visibox/error <message:string>
```

You get one for an unknown Control letter, a missing value, or a value sent as a string. Three things to know before you rely on it:

- **It needs somewhere to land.** The reply goes to a feedback endpoint subscribed from the same IP address, and otherwise back to the port the message came from. Many OSC apps send from a port they never listen on, so unless you subscribe first, the reply is thrown away.
- **Replies are matched by IP address**, not by application. If two OSC apps are running on one machine and one of them is subscribed, it gets the other's error messages.
- **Replies are throttled** to one per sender per second. A fader sends messages far faster than that, and one error you read beats two hundred you drop.

Examples

TouchOSC

1. Open TouchOSC on your phone or tablet.
2. Set the target host to your computer's IP address.
3. Set the target port to 17735.
4. Give each button and fader an OSC address from the tables above.
5. For feedback, set a receive port in TouchOSC, then send /visibox/subscribe with that port number.

A fader mapped to /visibox/control/A sending a float from 0 to 1 is the whole setup for driving a Control from a tablet.

QLab

Use a Network cue. Set its destination to your computer's IP address and port 17735, then set the OSC message on the cue. A cue that reads /visibox/clip/trigger 3 fires the fourth Clip of the current Song as part of your cue stack, so lighting, sound, and visuals all move on one Go.

Resolume

Resolume can send OSC out to another application. Point its OSC output at your computer's IP address and port 17735, then map a Resolume control to a Visibox address. Sending /visibox/song/next from a Resolume clip lets one operator's deck advance the other's.

Max

```
[udpsend <your-ip> 17735]
|
[prepend /visibox/play]
|
[bang]
```

Swap the address in the prepend object for any of the ones above. To drive a Control, put a float in front of the address:

```
[udpsend <your-ip> 17735]
|
[prepend /visibox/control/A]
|
[slider]
```

From a terminal

The oscsend tool that ships with liblo is a quick way to test an address without building anything:

```
oscsend localhost 17735 /visibox/play
oscsend localhost 17735 /visibox/clip/trigger i 0
oscsend localhost 17735 /visibox/volume f 0.75
oscsend localhost 17735 /visibox/control/A f 0.75
oscsend localhost 17735 /visibox/subscribe i 9000
```

Troubleshooting

Nothing happens at all. The sender is probably not paired. Pair the device in [Settings > Remote](#) first, even if it is on the same machine.

Playback addresses work but Control addresses do not. Real-Time Control is answered separately. Subscribe for feedback and read `/visibox/error` to see why a message was refused.

A fader does nothing until it reaches a certain point. That is takeover on the binding, doing its job. See [Real-Time Control](#).

The wrong clip fires, always one off. Send `/visibox/config/index-base 1` from your surface, or renumber your buttons from zero.

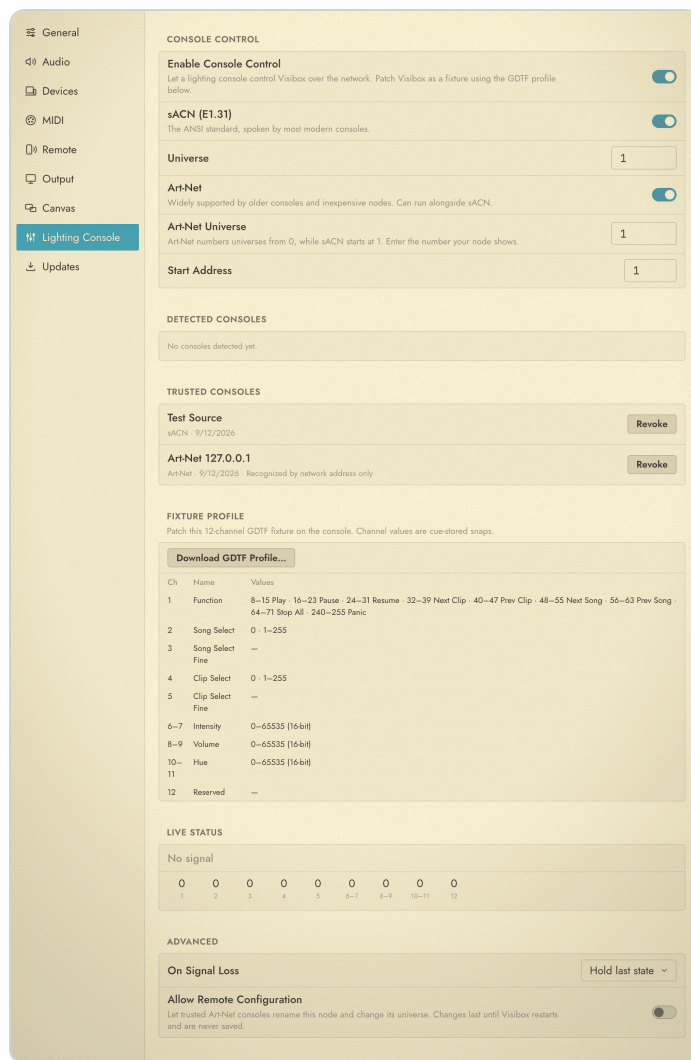
See Also

- [Real-Time Control](#) for what Controls are and how to shape them.
- [Remote Pairing](#) for pairing a device.
- [Visibox API](#) for the WebSocket and HTTP interfaces.

Lighting Consoles

A lighting console can run Visibox the way it runs a moving light. Patch Visibox as a fixture, and cues on the desk select Songs, fire Clips, ride the master intensity, volume and hue, and stop everything on a Go. The show itself stays authored in Visibox: which Clips loop, how they fade, what Effects they carry. The console picks and triggers what you built.

Visibox speaks the two protocols consoles use to send DMX over Ethernet, **sACN** (ANSI E1.31) and **Art-Net**. Either works alone, and they can run together. There is no DMX port to wire. The console and the computer running Visibox only need to share a network.



Settings > Lighting Console.

Setting up

1. Open Settings and choose **Lighting Console**. Turn on **Enable Console Control**.
2. Turn on the protocol your desk sends. **sACN (E1.31)** is on by default and is what most modern consoles speak. **Art-Net** is what older desks and inexpensive nodes speak. Turn on both if you are not sure. A console that sends either one will be heard.

3. Set the **Universe** and **Start Address** to match where you patch Visibox on the console. Both start at 1, and Visibox occupies twelve channels from the start address. With Art-Net on there is an **Art-Net Universe** field as well. See [Universe numbers](#).
4. Click **Download GDTF Profile...** and import the file into your console, then patch one fixture of it. grandMA3 imports GDTF directly. Most other desks have a GDTF import or a converter for it.
5. Send DMX from the console. The first time an unfamiliar console sends data, Visibox asks whether to allow it. Click **Allow**, and that console stays trusted until you revoke it.

Consoles that announce themselves on the network appear under **Detected Consoles** as soon as Visibox is listening, each with a **Use universe** button that sets the universe to match. Consoles you have allowed are listed under **Trusted Consoles**, each with a **Revoke** button.

Live Status at the bottom of the panel shows whether Visibox is receiving, the packet rate, the value arriving on each channel, and the last action a cue fired. When you are wondering whether the console is getting through, look there first.

Universe numbers

sACN numbers universes from 1. Art-Net numbers them from 0. Visibox keeps a field for each protocol so you never have to convert between the two: type into each one the number that protocol's own console or node shows you. **Art-Net Universe** accepts 0, and the two fields do not have to hold the same number.

The fixture

The GDTF profile is a single fixture with one mode, **Visibox v1**, twelve channels long. One patched fixture is the whole Output. There are no layers to patch separately.

Channel	Name	What it does
1	Function	Transport. Each range of values is one command, listed below.
2	Song Select	Selects a Song by its position in the running order: 1 is the first Song, 2 the second. 0 does nothing.
3	Song Select Fine	Reserved.
4	Clip Select	Plays a Clip of the selected Song by its position. 0 stops.
5	Clip Select Fine	Reserved.
6 to 7	Intensity	The video dimmer, 16-bit. Full is the Output as you built it and 0 is black. Audio is unaffected.
8 to 9	Volume	The master audio level, 16-bit.
10 to 11	Hue	Rotates the color of the whole Output, Clips and Overlays alike, through 360 degrees. 16-bit. 0 leaves the picture untouched.

Channel	Name	What it does
12	Reserved	Keeps the footprint stable.

Songs and Clips are addressed by their position in the running order, the same numbers the Controller shows. Reordering rennumbers them, so lock the show order before you program the desk. A value can reach 255 Songs, and 255 Clips in a Song.

Function ranges

Channel 1 fires a command once, when its value enters a range. Holding the value there does nothing more. To fire the same command again, take the channel back to 0 first, so program cues as the value and then 0.

Value	Command
0 to 7	Idle. Re-arms the channel.
8 to 15	Play
16 to 23	Pause
24 to 31	Resume
32 to 39	Next Clip
40 to 47	Previous Clip
48 to 55	Next Song
56 to 63	Previous Song
64 to 71	Stop All
240 to 255	Panic. Reloads every Visibox window. It can only be reached from Idle, so a fader ridden up the channel cannot hit it.

The values in between are reserved. Passing through them disarms the channel, so go back to 0 before the next command.

Intensity is a blackout, not a stop

Intensity at 0 takes the picture to black, Clips and Overlays alike, and leaves playback running underneath. Bring it back up and the picture returns instantly, still in time with the music. The audio keeps playing throughout, so pull Volume down as well if you want silence.

Stop All ends playback. Bring the look back and it starts over. Use Intensity when you want the screen gone for a moment and back where the music is. Use Stop All to end a look.

The ends of the Intensity range are instant, so a cue that snaps to 0 goes black on the frame rather than over a fade. Values in between are smoothed, so a fade from the desk ramps rather than steps. Volume and Hue are smoothed the same way. That is why all three are 16-bit: the console performs the fade, and the fine byte keeps the bottom of it from stepping.

Cues, not parked faders

Visibox has a live operator at the keyboard, and the console never overrules them. A value the console sends is acted on once, when it changes. A fader parked on Clip 5 does not keep re-asserting Clip 5, so the operator can stop that Clip and it stays stopped until the desk sends something new. Whoever acted last, console or operator, wins.

Program the desk in cues instead of leaving faders up: select the Song, select the Clip, then Go. When Visibox first hears a console, on enable, when the console boots, or when its signal returns, it takes in the current values silently and acts on the next change.

One console controls Visibox at a time. When two send sACN on the same universe, the one with the higher sACN priority wins. Art-Net has no priority, so the first console heard wins. Powering on a backup desk beside the main one does not disturb the show.

When the signal drops

If no data arrives for a few seconds, or the console ends its stream, **Live Status** reads **Signal lost**. What happens next is the **On Signal Loss** setting under **Advanced**.

- **Hold last state**, the default, leaves everything as it was.
- **Stop playback** stops the Output and restores full intensity.

When the console comes back, Visibox picks up its current values without replaying any cues.

Trusting a console

A console becomes trusted the first time it sends DMX to Visibox and you click **Allow**. Visibox remembers it by the strongest identity the protocol offers. An sACN console carries a permanent ID that survives reboots and address changes. An Art-Net console is remembered by its hardware address, or, when a node does not report one, by its network address alone. Those read **Recognized by network address only** in the list, and if the address changes, Visibox asks again.

Revoke next to a trusted console forgets it. The next time it sends DMX, Visibox asks again.

Letting the desk configure Visibox

Art-Net can rename a node and change the universe it listens on from the console. Visibox allows this only when **Allow Remote Configuration** under **Advanced** is on, and only from a console you have already trusted. Changes made this way last until Visibox restarts and are never saved, and the panel names the console that made them. Leave it off unless you want to re-patch Visibox from the desk.

Troubleshooting

Visibox is not in the console's node list. Only Art-Net announces Visibox by name. Turn Art-Net on even if the cues arrive over sACN. The two work together: Art-Net for finding and managing, sACN for the data.

Live Status says No signal. Check that the console is sending to the universe the panel shows, and that both machines are on the same network. Over Art-Net, check the **Art-Net Universe** field rather than **Universe**, since the two protocols number universes differently.

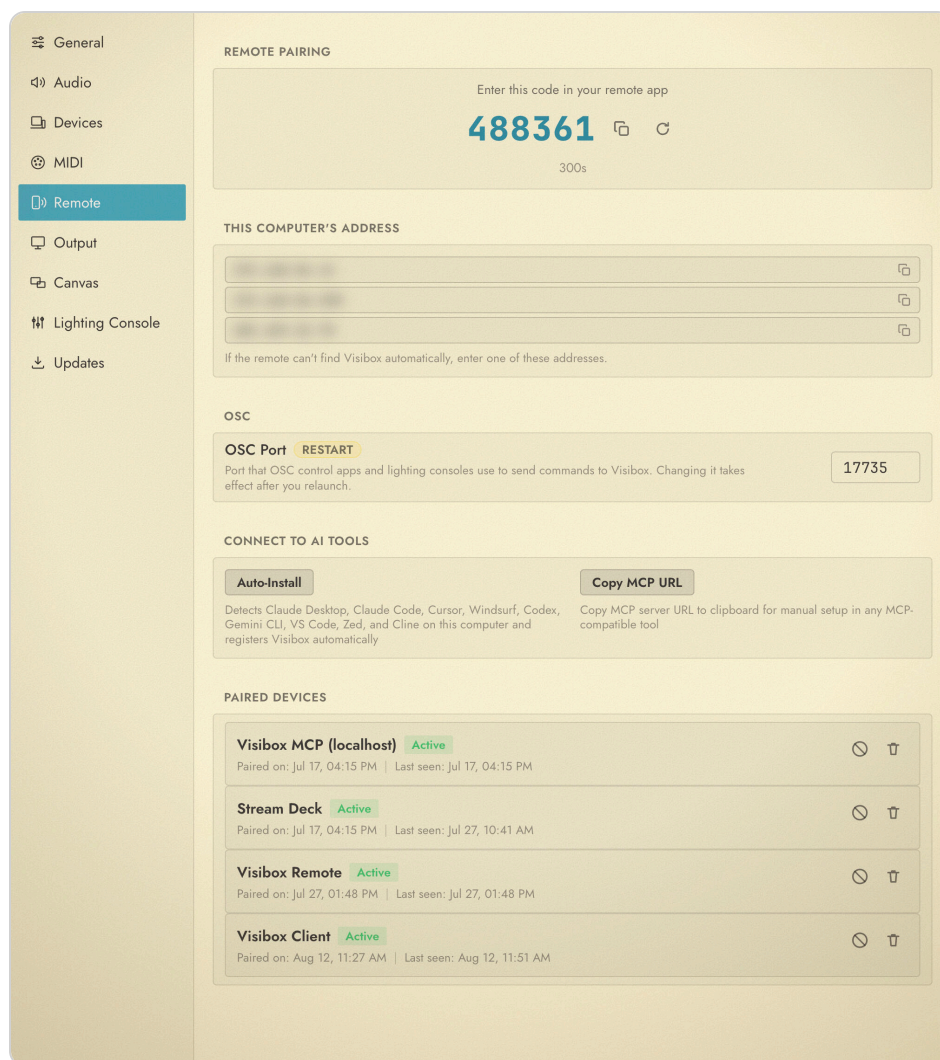
A cue selects the wrong Song or Clip. The running order has changed since the desk was programmed. Check the numbers in the Controller against the values in the cue.

The trust prompt keeps coming back. The console is recognized by network address only, and that address is changing. Give it a fixed address.

Visibox also answers to [OSC](#), which many consoles can send alongside DMX. Use OSC for anything the fixture does not cover, such as triggering a Clip by name.

Remote Pairing

The Remote category of the [Settings window](#) is where you connect external devices and AI tools to Visibox over the network. Choose **Pair with Remote...** to open Settings on that category (on Mac, from the Visibox menu; on Windows, from the File menu), or open Settings and pick **Remote** in its sidebar. In Visibox 5 this was a window of its own.



Settings > Remote.

Pairing a Device

Devices connecting to Visibox over the network must pair first using a one-time code:

1. Choose **Pair with Remote...** to open Settings on the Remote category.
2. A 6-digit code appears under **Remote Pairing** with a countdown timer. The code expires after 5 minutes.
3. On the remote device, enter the code (or send it via the [API](#)).
4. Visibox returns a token that the device stores for future connections.

If the code expires, click **Generate new code** to get a fresh one. **Copy code** puts the current code on the clipboard.

When the remote app cannot find Visibox on its own, a **This computer's address** section lists the addresses to type into the remote by hand. It appears only when Visibox has a network address to show.

The first time a device on the same machine connects (a Stream Deck plugin, for example), Visibox asks you to allow it. That prompt always opens over a window you can see, so it will not go missing behind or off the edge of your screens.

NOTE

A program on the same machine never needs a pairing code. How it gets in depends on how it connects. The [MCP server](#) and the [command line tool](#) reach Visibox over HTTP and are let through with no token and nothing to approve. Something that holds a live connection, like the Stream Deck plugin, gets the approval prompt above once, and Visibox remembers it after that.

Token Lifetime

A pairing token lasts 30 days. Every time a paired device connects, Visibox issues it a fresh 30-day token, so a device you use regularly stays paired with nothing to renew and nothing to set up. That covers Visibox Remote, the Stream Deck plugin, the [command line tool](#) when it connects over the network, and any device you paired yourself.

A device that sits unused for more than 30 days has to pair again with a new code. If you paired a device before updating to Visibox 6.0, it keeps working and picks up the new token the next time it connects.

Connecting AI Tools

The **Connect to AI Tools** section of Settings > Remote can install the Visibox MCP server configuration into supported AI tools for you. Click the **Auto-Install** button to detect and configure any of them. See [AI Assistants](#) for the full list.

You can also click **Copy MCP URL** to manually add the Visibox MCP server to any tool that supports the Model Context Protocol. The same scan runs from the **Connect to AI Tools...** menu item without opening Settings. See [AI Assistants](#) for more about what AI tools can do with Visibox.

OSC Port

The **OSC** section holds the port Visibox listens on for [OSC](#) commands, from control apps and from lighting consoles. It is 17735 unless you change it, and it accepts any port from 1024 to 65535. A new port takes effect after you relaunch Visibox.

Managing Paired Devices

The **Paired Devices** section at the bottom of Settings > Remote lists all paired devices with their status:

- **Device name.** Identifies the paired device or tool.
- **Status.** Shows whether the device is **Active** or **Revoked**.
- **Paired on.** When the device was first paired.
- **Last seen.** When the device last connected. This is also when its token was last refreshed.

You can **Revoke access** to disconnect a device, or **Remove device** to take it off the list entirely. A revoked device has to pair again with a new code.

Editors

Visibox has two kinds of editor, and they do different jobs.

- **Editor Panels** are the popups you get by right-clicking something in the Controller window. They edit one Clip, Song, or Project at a time, and they close when you are done.
- **The Editor Window** is a tabbed window holding the Effects, Visualizers, and [Overlays](#) editors. It is where you build the presets that the panels then assign.

Reach for a panel to change this Clip. Reach for the Editor Window to change the preset every Clip uses.

Editor Panels

Right-click anything in the Controller window to open its panel:

- **Right-click a Clip** to open the [Clip Editor Panel](#).
- **Right-click a Song** to open the [Song Editor Panel](#).
- **Right-click the background** of the Song list to open the Edit Project panel. See [The Visibox Interface](#).

The panel opens at your cursor with the options that apply to what you clicked. It works from the keyboard too:

- Arrow keys move through the rows and open or close their sub-panels.
- Typing jumps to a matching row.
- Escape steps back out.

Editing several items at once

Select several Clips or Songs first, then right-click any one of them. The panel opens on the whole selection.

To build a selection:

- Hold  (Shift) and click to select a range.
- Hold  and click to add or remove one item.
- Drag a box across the Clip grid to select everything it touches.

The panel shows the options common to everything you selected. Where a value differs across the selection, it reads **Mixed**. Change it and every selected item takes the new value.

For more on selecting Clips, see [Adding Clips](#). For Songs, see [Adding Songs](#).

Subpanels

Some rows open a **subpanel** to the side when you hover over them:

- **Sliders** for continuous values like volume, speed, and fade times. Click the displayed value to type an exact number.
- **Radio lists** for picking one value, such as Fit Mode or Launch Mode.

- **The filter editor** for turning several visual filters on and adjusting each one.
- **The color picker** for choosing a color from the system picker, a hex value, or a preset swatch.
- **Tweak panels** for the parameters a Clip's Effect or Visualizer preset exposes. See [Effects](#).

Double-click a slider to put it back to its default.

The Editor Window

Effects, Visualizers, and Overlays each have a full editor, and all three live as tabs in one window. See [The Editor Window](#) for how to open it and work with tabs.

Panels and menus

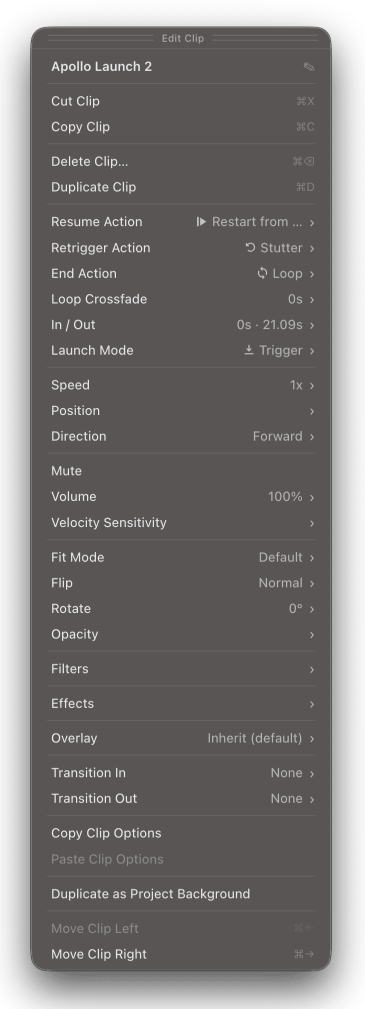
Most Clip and Song options also appear in the [Clip Menu](#) and [Song Menu](#) in the menu bar. The panels give you finer control:

- **Continuous sliders** instead of preset steps. Set speed to 1.37x, not only 1x or 2x.
- **Typed values.** Click any slider value and type an exact number, even past the end of the slider.
- **A full color picker** instead of a short list of colors.
- **Visual filter editing.** Enable, adjust, and combine filters with their own sliders.

The menus are quicker for a single known change. The panels are better for dialing something in.

Clip Editor Panel

Right-click any Clip thumbnail in the Controller to open the Clip Editor Panel, a floating panel holding every setting a Clip has. **Clip > Edit Clip...** opens the same panel.



The Clip Editor Panel for a Video Clip.

Which rows appear depends on the kind of Clip you right-clicked. A Video Clip gets the playback and audio rows, a Visualizer Clip gets rendering rows, a Color Clip gets a color picker. A row that makes no sense for the kind of Clip is left out. A row that another setting is currently blocking stays in place, grayed, showing the value in effect.

Select several Clips first and the panel edits everything they have in common. Where the selected Clips disagree on a value, the row reads **Mixed** until you set it.

Section	Video	Audio	Image	Camera	Visualizer	Color
Playback	Yes	Yes	Yes	Yes	Yes	Yes
Audio	Yes	Yes		Yes		

Section	Video	Audio	Image	Camera	Visualizer	Color
Display	Yes	Yes	Yes	Yes	Yes	Yes
Filters	Yes	Yes	Yes	Yes	Yes	Yes
Effects	Yes	Yes	Yes	Yes	Yes	Yes
Audio Source		Yes			Yes	
Overlay	Yes	Yes	Yes	Yes	Yes	Yes
Transitions	Yes	Yes	Yes	Yes	Yes	Yes
Visualizer		Yes			Yes	
Camera				Yes		
Actions	Yes	Yes	Yes	Yes	Yes	Yes

Two sections vary inside themselves. Under Playback, Video and Audio Clips get the trim rows and the resume, retrigger and end rows. Image, Camera and Visualizer Clips get a **Duration** instead, and until that Duration is set, their Resume Action row is left out altogether and their End Action row is grayed. Display varies too: see [Display](#) below for which Clip kinds skip Fit Mode, Flip or Rotate.

An Audio Clip follows whichever it currently resembles. On its own it is a color, so it gets the color picker and none of Fit Mode, Flip or Rotate. Pair it with a Visualizer and there is a picture to place, so all three appear. It also loses the rows that lean on video-only machinery:

- **Direction** is gone entirely.
- **End Action** has no Bounce.
- **Retrigger Action** has no Reverse or Scratch.
- **Resume Action** has no Sync to Song Position.
- **Loop Crossfade** is gone.

See [Audio Clips](#).

Audio Source follows its own availability rule, covered in [Audio Source](#) below.

Renaming and clipboard

Rename

The top row of the panel is the Clip's name. Click it and type to rename the Clip. It appears only when a single Clip is selected, since a group of Clips has no one name to edit.

Cut Clip ⌘X

Copy Clip ⌘C

Cut or copy the selected Clips. A cut Clip stays in place, outlined and dimmed, until you paste it into a Song. The labels count the selection, so they read **Cut 3 Clips** and **Copy 3 Clips**. Paste with **⌘V** after selecting the Song you want them in.

Playback

A Color Clip carries only **Launch Mode** here. It has no length, no trim and no end to reach, so nothing else in this section applies to it.

Resume Action ›

Where the Clip picks up when you return to it after playing something else.

- **Restart from Beginning** starts it over every time.
- **Resume from Last Point** continues where it stopped.
- **Sync to Song Position** locks the Clip to the Song's audio timeline. See [Sync to Song Position](#).

NOTE

Sync to Song Position restricts some of the rows below. Retrigger Action, End Action and Direction each fall back to the value sync allows, and Loop Crossfade is grayed. The panel labels those rows **Restricted in Sync Mode**.

Retrigger Action ›

What a press on a Clip that is already playing does. Restart is the default. The full list and its thumbnail icons are on [Clip Playback Options](#).

A Clip whose [Launch Mode](#) is Toggle stops on its second press instead. This row is grayed for such a Clip and shows **Stop** as the value in effect.

End Action ›

What happens when the Clip runs out: **Loop**, **Bounce**, **Stop (One-Shot)**, **Freeze**, **Start Next Clip**, **Cue Next Song** or **Start Next Song**. See [End Action](#).

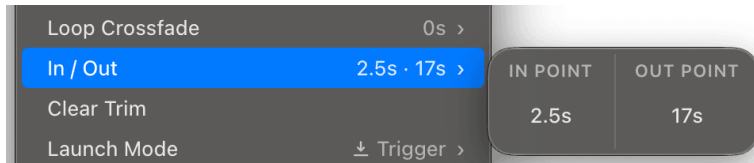
Loop Crossfade ›

Blends the end of a looping Clip into its start so the wrap point stops reading as a cut. Set a duration in seconds; a second or two usually does it.

The row appears only when End Action is set to Loop. It is grayed for a Clip playing backward or set to Sync to Song Position, since neither can use a crossfaded loop. Until the crossfaded version is ready, the Clip loops with a hard cut as it always did.

In / Out ›

Pick the section of the file a Video Clip plays. The row shows both times side by side, and its panel holds an **In Point** field and an **Out Point** field. Click a value to type an exact time, down to a hundredth of a second. A point you have not set reads the natural edge of the file: the start for In Point, the end of the media for Out Point.



In Point and Out Point choose the section of the file the Clip plays.

Set In Point at Playhead

Set Out Point at Playhead

Set a trim point to wherever the Clip is right now. Both rows appear only while a single Clip is playing or paused, and both act the moment you press them.

Clear Trim

Removes both points and returns the Clip to the full file. The row appears once the Clip has a trim to clear.

Duration ›

For Image, Camera and Visualizer Clips, which have no length of their own. Leave it at ∞ and the Clip plays until you move on. Give it a positive value and the Clip ends after that many seconds, at which point End Action applies.

Launch Mode ›

What the trigger gesture does: **Trigger** starts the Clip, **Gate** plays it while you hold, **Toggle** starts it and stops it on a second press. See [Launch Modes](#).

Leave it unset and the Clip follows its Song: Toggle in a Song with a co-play [Poly Mode](#), Trigger in a Mono Song. The panel shows which mode is in effect either way.

Speed ›

Playback speed for a Video Clip, from 0.0625x to 4x, with snap points at 0.25x, 0.5x, 1x, 2x and 4x. Click the value to type an exact speed, including one past either end of the slider. Double-click the slider to return to 1x.

Position ›

Video Clips only. Assigns the Clip's playhead to one of the Project's [Controls](#), so a fader or knob can scrub the Clip like a record. The row carries no value of its own and appears only when there is a Control to assign it to.

Direction ›

Forward or **Backward**. A Clip playing backward is muted and cannot use a Loop Crossfade.

Audio

Video, Audio and Camera Clips. A Camera Clip gets Mute and Volume but not Velocity Sensitivity, since nobody triggers a live feed like a drum pad.

Mute

Silences the Clip's audio track.

Volume ›

The Clip's level. The slider runs from 0% to 200%, and you can click the value and type higher when a Clip is much quieter than everything around it. Double-click the slider to return to 100%.

Velocity Sensitivity ›

How far a MIDI note's velocity is allowed to move the Clip's audio level. At 0, the default, velocity is ignored and the Clip always plays at its Volume setting. Turn it up and a harder hit plays louder.

Triggers with no velocity of their own, such as the mouse, the keyboard, a Stream Deck, OSC and the API, play the Clip at full level whatever the setting.

With Retrigger Action set to Stutter, Scratch, Reverse or Do Nothing, only the first hit sets the level. The panel captions the slider **First hit only** when that applies. See [Velocity Sensitivity](#).

Display

A Color Clip gets the color picker but no Fit Mode, Flip or Rotate. Fitting, mirroring and turning a solid color would show nothing, so those rows are left out. A Visualizer Clip gets Flip and Rotate but no Fit Mode, since it renders to the frame rather than into it.

Color ›

Color Clips only. Choose the Clip's color three ways:

- Click the swatch to open your operating system's color picker.
- Type any CSS color value: a hex code such as `#FF5500`, a name such as `tomato`, or any other CSS color format. Press Enter to apply it.

- Click one of the twelve preset swatches: Black, White, Gray, Red, Orange, Gold, Yellow, Green, Sky Blue, Blue, Purple and Magenta.



TIP

The picker takes exact values, which the [Clip menu](#) preset list cannot. Use the panel when you need a specific color.

Fit Mode ›

How the Clip fills the Output: **Default** (the Project's setting), **Crop**, **Letterbox** or **Stretch**. See [Fit Mode](#).

Flip ›

Mirrors the Clip: **Normal**, **Horizontal**, **Vertical** or **Both**.

Rotate ›

Turns the Clip to any angle. The slider runs from 0 to 360 degrees and snaps at 0, 90, 180 and 270, so square turns are easy to hit and everything between them is available. Click the value to type an exact angle.

Opacity ›

Assigns the Clip's opacity to one of the Project's [Controls](#), so a fader can ride it live. It carries no value of its own and appears only when there is a Control to assign it to. Unlike Position, it is offered on every kind of Clip.

Filters

Filters ›

Image adjustments applied live, so the original file is untouched. Each filter has a switch and a slider:

1. Turn a filter on and its slider appears.
2. Drag the slider to set the amount.
3. Click the value to type an exact number.
4. Double-click the slider to reset that filter.
5. **Reset** at the bottom clears all of them.

Filter	Range	What it does
Blur	0–10%	Softens the image
Brightness	0–5	Brightens or darkens
Contrast	0–5	Increases or reduces contrast

Filter	Range	What it does
Grayscale	0–1	Desaturates toward black and white
Rotate Hue	0–360°	Shifts every color around the color wheel
Invert	0–1	Inverts the colors
Saturate	0–5	Boosts or reduces color
Sepia	0–1	Applies a warm, vintage tone

Each filter can also be assigned to a [Control](#), so a fader can ride it during a show.

Effects

Effects ›

Apply visual Effects to the Clip. The panel lists the Effect library by category. Pick one to apply it, stack several to layer them, and use **Reset** to clear them all.

Edit Effects... at the top opens the Effects Editor, where you can browse, customize and build Effects. See [Effects](#).

Tweak ›

Adjusts the parameters the Clip's Effects declare, for this Clip only. The row sits directly under Effects and appears only once an Effect on the Clip has something to tune, so there is never an empty panel behind it. The count next to the label is how many parameters you have changed.

A Visualizer Clip gets its own Tweak row under the Visualizer row, covering that preset's parameters. See [Effects](#).

Overlay

Overlay ›

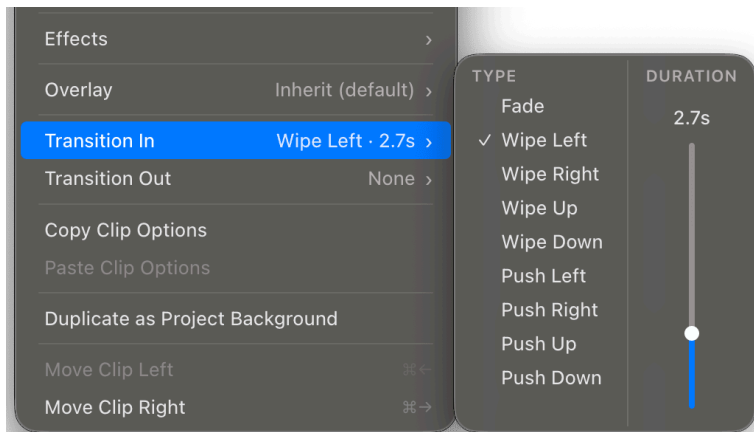
Assigns an [Overlay](#), such as a logo or a lower third, to show above this Clip. A Clip's Overlay takes over from the Song's and the Project's.

Transitions

Transition In ›

Transition Out ›

How the Clip arrives on stage and how it leaves. Each row carries a shape and a duration together: **Fade**, **Wipe** in four directions, or **Push** in four directions, plus a duration in seconds.



A transition sets its shape and its duration in one place.

A duration of zero is a cut whatever shape is chosen, and the row reads **None**. Transition In also sets the crossfade from the previous Clip, and takes precedence over that Clip's Transition Out. See [Transitions](#).

Visualizer

Visualizer Clips only.

Visualizer ›

Choose the preset. The panel lists the available MilkDrop and ISF presets with thumbnails. **Edit Visualizers...** opens the Visualizers Editor. See [Visualizers](#).

Tweak ›

Adjusts the chosen preset's own parameters for this Clip. It appears only for a preset that declares parameters, so a MilkDrop preset shows no row.

Resolution ›

The rendering resolution, from 0.25x to 2x. Lower is faster; higher is sharper.

Target FPS ›

The frame rate to aim for, from 15 to 120, with snap points at 15, 30, 60 and 120.

Audio Source

Audio Source ›

What the Clip's audio-reactive rendering listens to:

- **None** turns audio reactivity off.

- **This Clip** reacts to this Clip's own sound and nothing else. Audio Clips only, and only once a Visualizer is paired with one. It reads the level after the Clip's Volume, so turning the Clip down calms the picture with it, and two Audio Clips playing at once each drive their own.
- **Visibox** reacts to whatever Visibox is playing, Song audio and Clip audio alike.
- **System Audio** reacts to everything coming out of the computer.
- Any connected audio input, listed below the separator, reacts to a microphone or interface.

Visualizer Clips always have this row. An Audio Clip gets it once a Visualizer is paired with it. Any other Clip gets it once it carries an Effect, since an Effect is the other thing that can react to audio.

A new Audio Clip starts on **This Clip**. Everything else starts on **Visibox**.

Camera

Camera Clips only.

Camera ›

Choose the connected camera or video input this Clip shows. See [Connecting Cameras & Video Inputs](#).

Actions

Copy Clip Options

Paste Clip Options

Copy captures one Clip's settings, without its media. Paste applies them to every Clip in the current selection. This is the fast way to make a whole Song behave alike once you have one Clip right.

Delete Clip... ⌘⌘

Deletes the selected Clips. The label counts them, so it reads **Delete 3 Clips....**

Duplicate Clip ⌘D

Copies the selected Clips. Duplicates share one media file on disk, so they cost no extra space.

Duplicate as Project Background

Copies the Clip and makes it the Project's Background Clip, the one that plays whenever nothing else is up.

Move Clip Left ⌘←

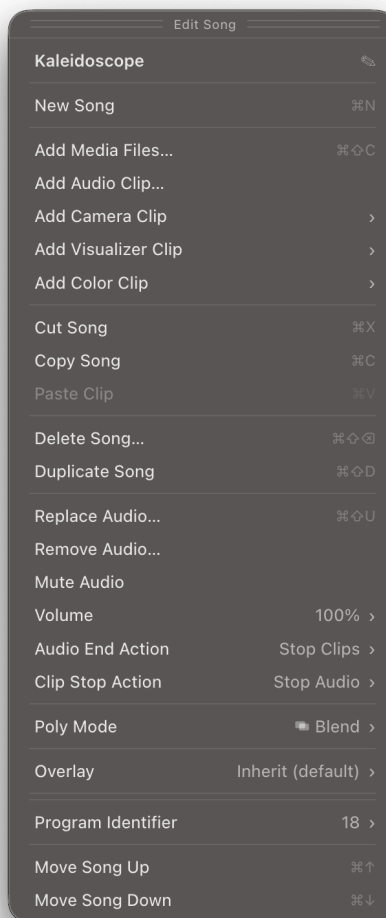
Move Clip Right ⌘→

Moves the selected Clips within the Song. Dragging them does the same thing.

Song Editor Panel

The Song Editor Panel is the full set of controls for a Song. Open it three ways:

- Right-click a Song in the Controller window.
- Choose **Song > Edit Song...** from the menu bar.
- Click the chevron in a Song row's badge cluster.



The Song Editor Panel. Right-click any Song to open it.

The menu-bar [Song menu](#) carries the same actions and the settings that fit a list of choices. What lives only here is what a menu cannot draw: the audio volume slider, the Overlay picker, and the freeform color picker behind Add Color Clip.

Some rows appear only when a single Song is selected. With several Songs selected, the panel edits the settings they have in common and shows a mixed marker where their values differ.

Adding Clips

These rows add new Clips to the Song. They appear when one Song is selected.

Add Media Files... C

Open a file dialog and pick one or more video, image, or audio files to add to the Song. Same result as dragging files from the desktop onto the Song.

Add Audio Clip...

Pick one or more audio files and add each as its own Clip. See [Audio Clips](#).

Add Camera Clip

Add a live video feed as a Clip. The subpanel lists every connected camera and video input. Pick one to add it. See [Connecting Cameras & Video Inputs](#).

Add Visualizer Clip

Add a Visualizer Clip: live, generative, audio-reactive visuals. The subpanel lists your Visualizer presets. **Edit Visualizers...** at the bottom opens the Visualizers editor. See [Visualizers](#).

Add Color Clip

Add a solid-color Clip. The subpanel offers a set of preset swatches, from black, white and gray through the warm and cool colors.

TIP

A Song needs a Clip playing before it will play Song audio, so a black Color Clip is the clean way to play “nothing” on screen. You can also stack [Effects](#) on a Color Clip to build a moving image out of a flat color.

TIP

For a color the swatches do not cover, add a Color Clip and then right-click it. The [Clip Editor Panel](#) has the full picker with hex values.

Clipboard

Cut Song ⌘X

Copy Song ⌘C

Put the selected Songs on the clipboard, with their Clips and settings. Cut removes them from the Project at the same time. With more than one Song selected the rows count them: **Cut 3 Songs, Copy 2 Songs**.

Paste into another open Project and the media files travel with the Songs.

Paste Clip ⌘V

Paste Clips from the clipboard into this Song. The row counts what it will paste (**Paste Clip, Paste 4 Clips**) and is unavailable when the clipboard holds no Clips.

Audio

These rows manage the Song's backing track: audio that plays alongside the Song's Clips.

Attach Audio... / Replace Audio... ↑ ⌘U

Open a file dialog and pick an audio file. Once a Song has audio, the row reads **Replace Audio...**

Supported formats: MP3, AAC, M4A, WAV, FLAC, OGA, and OPUS.

Remove Audio...

Take the audio track off the Song. Available only when the Song has one.

Mute Audio

Silence the backing track without removing it.

Volume ›

Set the backing track's level, from 0% to 200%. Click the displayed value to type an exact percentage.

Audio End Action ›

What happens when the backing track finishes:

- **Stop Clips** stops the playing Clip.
- **Clips Keep Playing** leaves the Clips running.
- **Loop Audio** starts the track again from the top.
- **Cue Next Song** moves to the next Song and waits.
- **Start Next Song** moves to the next Song and starts it.

Clip Stop Action ›

What happens to the backing track when the last playing Clip stops. That covers letting go of a Gate Clip, switching a Toggle Clip off, or stopping it any other way:

- **Stop Audio** stops the track with the Clip.
- **Audio Keeps Playing** leaves it running, so the next Clip you fire lands on a track that never stopped.

A Song you have not set follows its Poly Mode. A Mono Song stops the audio, as Visibox 5 did. A polyoptic Song keeps it playing, because its Clips are usually voices played over the track. The row shows the value in effect either way.

Playback

Poly Mode ›

Poly Mode decides how this Song puts more than one Clip on screen at once. The subpanel offers **Mono** (one Clip at a time, the default) plus six ways to share the stage: **Blend**, **Mosaic**, **Horizontal Stripes**, **Vertical Stripes**, **Stack**, and **Spots**.

The [Poly Mode](#) page covers what each one looks like and how the app-wide limit on simultaneous Clips works. This panel is where you set it per Song.

Every Song carries a poly mark in its Controller row, Mono included. The mark shows which mode the Song is in, so you can read it without opening the panel.

Overlay ›

Assign an Overlay to the Song. Everything the Song plays then shows that logo or lower third on top.

The subpanel lists:

- **Inherit** takes whatever the Project assigns. The row names what that resolves to, so you do not have to go looking.
- **None** shows no Overlay for this Song, even when the Project assigns one.
- **Each of your Overlays**, one row per Overlay. Check more than one to stack them.
- **Edit Overlays...** opens the [Overlays](#) editor.

The panel assigns an Overlay. Position, size, margin, and opacity belong to the Overlay itself and are set in the Overlays editor, next to a live preview.

Song management

Edit Song Title...

Put the Song title into edit mode. Double-clicking the title in the Controller window does the same. One Song only.

Program Identifier >

Assign a MIDI Program Change number (1 to 128, or 0 to 127 if you have switched the numbering in Settings > MIDI) to the Song, so an external device or piece of software can select it by sending that Program Change. Choose **None** to clear it. One Song only.

See [Selecting Songs via MIDI Program Change](#).

Actions

New Song ⌘N

Add a new, empty Song to the Project.

Duplicate Song ⌘D

Copy the selected Songs. Duplicates point at the same media files as the originals.

Delete Song... ⌘⌘⌫

Delete the selected Songs and every Clip in them. With several selected the row counts them, as in **Delete 3 Songs....**

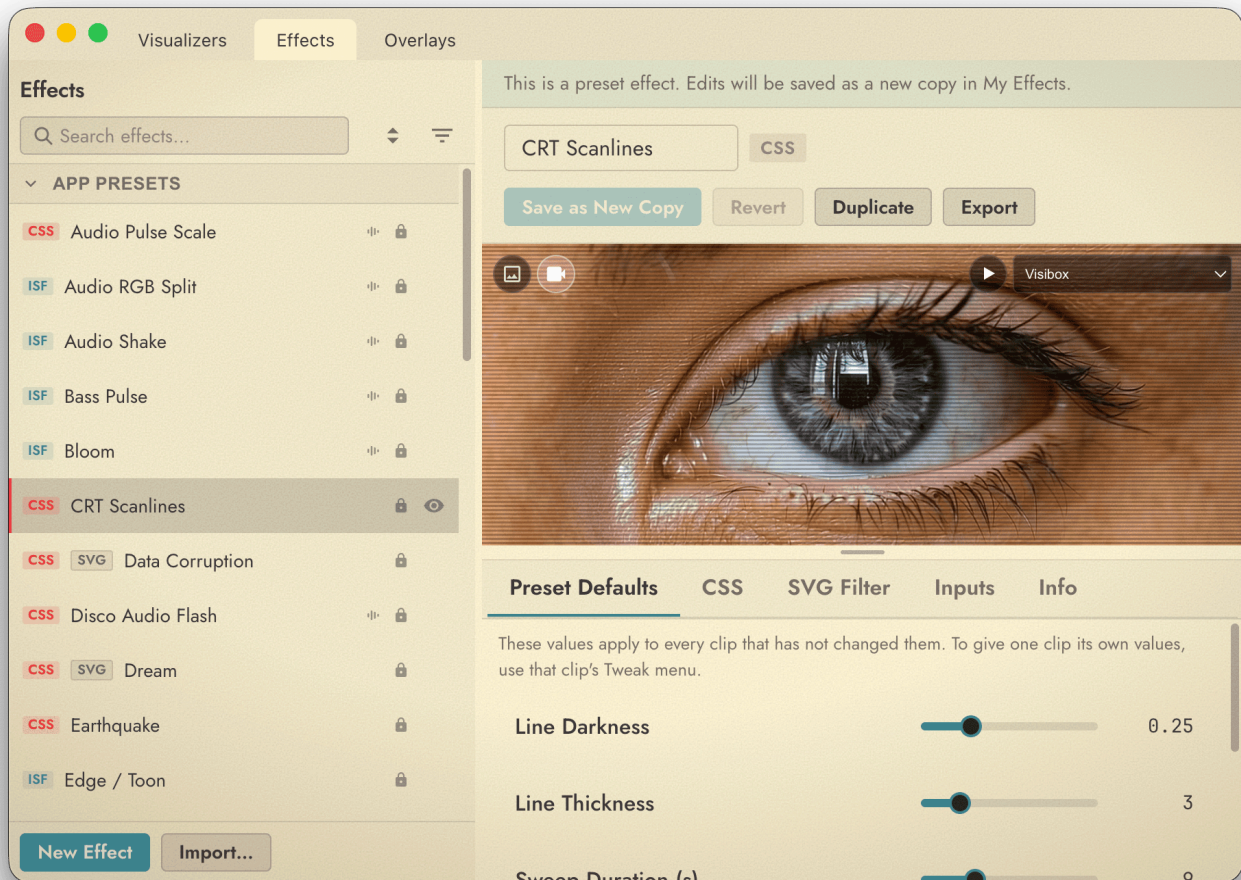
Move Song Up ⌘↑

Move Song Down ⌘↓

Move the selected Songs through the running order. Dragging a Song frame by its handle does the same. See [Setlists](#) for how the running order is stored.

The Editor Window

Effects, Visualizers, and [Overlays](#) share one window with a tab across the top for each. Open any one of the three and you get that window with that tab in front.



One window, three tabs. The window is named after whichever tab is in front.

Opening an editor

All three editors open from the Visibox menu on Mac and the File menu on Windows:

- **Effects...** () opens the Effects editor.
- **Visualizers...** () opens the Visualizers editor.
- **Overlays...** has no shortcut.

Each command brings its own tab to the front, opening the window first if it was closed. The Window menu also lists every open tab by name, so you can jump to one directly. See the [Window Menu](#).

What is in each tab

Tab	What it does
Effects	Browse, create, and edit CSS and ISF Effects. See Effects .
Visualizers	Browse, create, and edit MilkDrop and ISF Visualizer presets. See Visualizers .
Overlays	Build the logos and lower thirds that sit above everything the Output shows.

Each tab keeps its own state. Switching away from the Effects tab and back does not lose your place or reload the editor.

Working with tabs

Click a tab to switch to it. Drag a tab to change its order.

You are not stuck with one window. Drag a tab off the strip and drop it on empty screen space and it becomes a window of its own. Drag it back onto another window's tab strip and it rejoins that window, dropping in at the position you release it. A vertical bar shows where it will land.

You can split the three editors across up to three windows. This is worth doing on a large screen or a second display, when you are working on an Effect and an Overlay together. Each editor lives in exactly one window at a time. When you drag the last tab out of a window, that window closes.

Dragging a tab makes it the active one first, so what you see is what you are moving. Press  during a drag to cancel it.

The window remembers your layout

Visibox stores how you arranged the editors: which tabs are grouped together, their order, which one was in front, where each window sat, and whether it was open. Quit and relaunch and you get the same arrangement back.

The editors are not reopened until you ask for one. On a fresh install all three tabs start in one window with Visualizers in front, and that window stays closed until you open an editor for the first time.

If a display you had a window on is gone the next time you launch, Visibox moves that window back onto a screen you still have.

Song Timeline

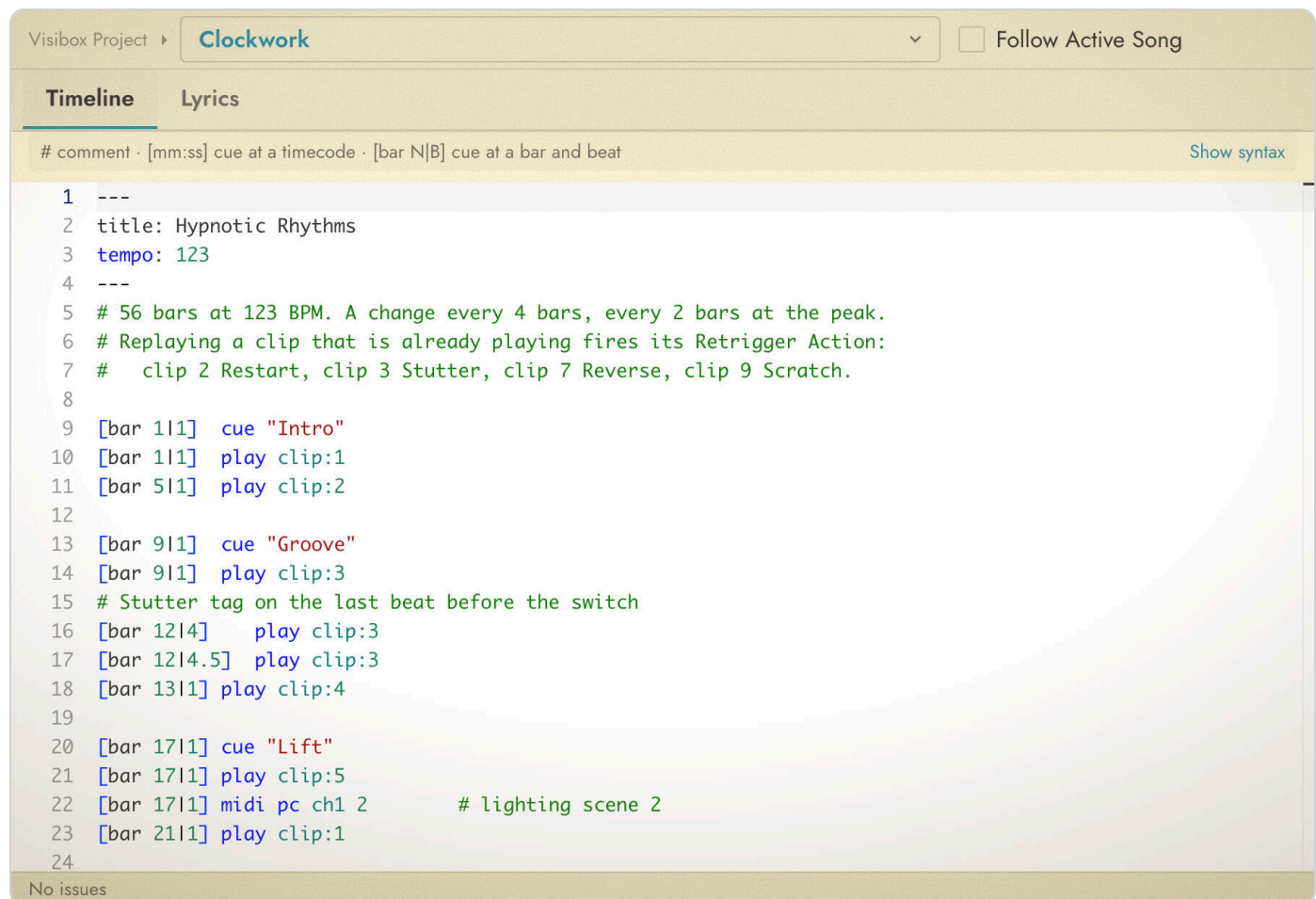
A **Song Timeline** is a list of cues attached to one Song. Each cue says when something should happen and what should happen. Visibox runs the list as the Song plays, so clips switch, the output level rides, and MIDI goes out to your rig without anyone touching the Controller.

Timelines suit a set you play the same way every night. If you want to stay hands-on, ignore them. A Song with no timeline behaves exactly as it always has.

Opening the Timeline Editor

Select a Song, then choose **Song > Edit Song Timeline... (Beta)**. The same row is in the Song's right-click menu.

Turn on Settings > General > Enable Beta Features to see this.



The Timeline Editor.

Across the top of the window is a breadcrumb showing which project and which Song you are editing. Both are pop-up menus, so you can move to another Song without closing the window. Next to them is a **Follow Active Song** checkbox. Turn it on and the editor re-binds itself as you move around the Controller, which is the mode you want while you are building a set.

The window has two tabs: **Timeline** and **Lyrics**. This page covers the Timeline tab. The Lyrics tab is covered on the [Lyrics](#) page.

Saving

There is no Save button. Visibox writes your edits back into the Project about half a second after you stop typing, and the status bar reads **Saving...** and then **✓ Saved** as it goes. Press  to save at once, and closing the window or moving to another Song saves immediately.

Because a pause in typing becomes one checkpoint rather than one per keystroke, **Edit > Undo** steps back through meaningful edits instead of single characters.

How a Timeline Runs

A timeline is anchored to the Song, and the Song starts when you trigger one of its clips. Every timecode in the list is measured from that moment. Cues written for a point you have already passed do nothing.

The timeline is an assistant, not an autopilot. You still call the top of the Song. Everything after it can be written down.

Writing Cues

Every line is one cue: a trigger in square brackets, then an action.

```
[00:24] play clip:2
[01:12] midi cc ch1 1 90
[on song end] next song
```

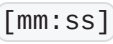
Lines beginning with  are comments and are ignored, and a  part-way through a line comments out the rest of it. Blank lines are fine. Cue order is free, because the trigger decides when a cue fires, not its place in the list.

A Song with no timeline yet opens with a starter document. Every line in it is commented out, so it does nothing until you delete a . It is the fastest way to see the shapes without leaving the window.

 **WARNING**

One bad line stops the whole timeline. Visibox will not run a document that contains an error, and the status bar says so: it reads **Timeline won't run until fixed** next to the error count. Check that the status bar is clean before you rely on a timeline in a show.

Triggers

Trigger	Fires
	At a point measured from the start of the Song.

Trigger	Fires
[mm:ss.fff]	The same, to the millisecond.
[90]	A plain number of seconds from the start of the Song.
[+secs]	A number of seconds after the previous timecode cue.
[bar B BT]	At a bar and beat, counted by the timeline's tempo. [bar 33 1] is the first beat of bar 33.
[on song start]	When the Song starts.
[on song stop]	When the Song is stopped.
[on song end]	When the Song reaches its end.
[on play clip:N]	When Clip N starts playing, whatever started it.
[on stop clip:N]	When Clip N stops.

The **on play** and **on stop** triggers are the useful ones for a set that does not run to a fixed clock. Write the reaction once and it fires whether the Clip was launched by the timeline, by a pad, or by your hand on the Controller.

[+secs] counts from the last timecode cue above it, so it needs one. A [+5] with no timecode cue before it is an error.

Clip numbers count from 1, in the order the Clips appear in the Song. You can also write a Clip's ID in place of the number.

Bar cues count by the timeline's tempo, set with the **tempo** action, usually at [00:00]. With no tempo set, bar cues fall back to 120 BPM in 4/4, and the editor points that out.

Playback actions

Action	What it does
play	Start playback.
play clip:N	Play Clip N in this Song.
stop clip:N	Stop Clip N.
stop all	Stop everything that is playing.
pause	Pause playback.
resume	Resume from a pause.
next clip / prev clip	Move to the next or previous Clip.

Action	What it does
<code>next song / prev song</code>	Move to the next or previous Song.
<code>activate song:N</code>	Select the Song in the current Setlist whose Program Identifier is N.
<code>activate song-id:<id></code>	Select a Song in the current Setlist by its ID.
<code>output-level 0-1</code>	Set the master output level, where 0 is off and 1 is full.
<code>tempo <bpm></code>	Set the tempo that [bar] cues count by, from this line on.
<code>panic</code>	Reload every Visibox window.

`activate song:N` matches the Program Identifier you set on the Song, not its position in the list. That is the same number a MIDI Program Change uses, so a timeline and a foot controller can call the same Song by the same number. See [Using MIDI](#).

Both `activate` forms look in the current Setlist. A Song that is not in the Setlist you are running cannot be called this way, and the editor flags the line as an error.

MIDI actions

A timeline can send MIDI out. This is how you drive a synth's patch, a lighting desk's cue, or a sampler hit from the same list that drives your visuals.

Action	What it sends
<code>midi pc ch<n> <program></code>	A Program Change on channel n.
<code>midi cc ch<n> <cc> <value></code>	A Control Change: controller number and value.
<code>midi noteOn ch<n> <pitch> <vel> <dur?></code>	A note on, with velocity. The duration in seconds is optional.
<code>midi noteOff ch<n> <pitch></code>	A note off.
<code>midi pitchBend ch<n> <value></code>	Pitch bend, 0 to 16383, centered at 8192.
<code>midi channelPressure ch<n> <pressure></code>	Channel aftertouch.
<code>midi polyPressure ch<n> <pitch> <press></code>	Aftertouch on one note.
<code>midi sysex <hex> <hex> ...</code>	A System Exclusive message. Start it F0 and end it F7.

The channel is written `ch1` through `ch16`, and the `ch` is required. Data values run 0 to 127. Pitches are written either as note names (`C3`, `F#2`, `Eb-1`) or as a plain number from 0 to 127. Note names follow the same numbering as the [MIDI Monitor](#), where `C3` is note 60. Keywords are case sensitive: `noteOn`, not `noteon`.

```
[00:00] midi pc ch1 12
[00:24] midi cc ch1 1 90
[01:12] midi noteOn ch2 C3 100 0.5
```

MIDI goes out of the default port unless you name one. To pick a port for a single cue, put `out:` and the port name in quotes straight after `midi`:

```
[00:00] midi out:"loopMIDI Port 1" pc ch1 12
```

See [Using MIDI](#) for setting up ports.

Syntax Help in the Window

The bar under the tabs shows a one-line reminder of what a cue looks like. Click **Show syntax** to open the full reference: the triggers, the playback actions, and the MIDI actions, in a grid you can leave open while you write. Click **Hide syntax** to close it again.

Errors and Warnings

Visibox checks the document as you type. Bad lines get a red underline and a marker in the left margin, and hovering one tells you what is wrong.

The status bar at the bottom reads **No issues** when everything parses. When it does not, it shows a count and the reminder that the timeline will not run. Click the count to jump the cursor to the first error.

Warnings are different from errors. A document with warnings and no errors still runs.

Jumping and Scrubbing

If you seek the Song forward, the cues you skipped over do not fire. Seeking backward re-arms them. Pausing cancels anything a cue had scheduled, and resuming picks up from where the Song is, without re-firing the Song's start reaction.

A Worked Example

A Song that runs itself from the timeline below:

- Sends a patch change to the keyboard rig on the downbeat.
- Moves to the band camera at the first chorus.
- Dims the output for the outro.
- Rolls into the next Song by itself.

```
# Opening
[00:00] midi pc ch1 12

# Chorus
[01:12] play clip:2
[01:12] midi cc ch1 1 90

# Outro
[03:40] play clip:3
[03:40] output-level 0.6

# Roll on
[on song end] next song
```

Editing a Timeline from Elsewhere

An AI assistant connected to Visibox can read and write a Song's timeline in plain language, which is often quicker than typing cues by hand. See [AI Assistants](#). Anything it writes appears in the Timeline Editor as it happens if the window is open, and **Edit** > **Undo** rolls it back.

Tips



TIP

Build the timeline with **Follow Active Song** on and the Controller beside the editor. Trigger the Song, watch it run, and fix the cue that landed wrong while the Song is still going.



NOTE

Comment a cue out with a `#` rather than deleting it. That is the quickest way to take one line out of a show without losing the wording you want back tomorrow.

See Also

- [Adding Songs](#) for attaching the Song audio a timeline runs against.
- [Using MIDI](#) for MIDI ports, Program Identifiers, and Program Change.
- [AI Assistants](#) for writing timelines in plain language.

Visibox Menu Tour

Keyboard shortcuts and right-click menus are usually the fastest way to work with Songs and Clips. Every shortcut also has a menu item at the top of the application, for when the mouse is easier.

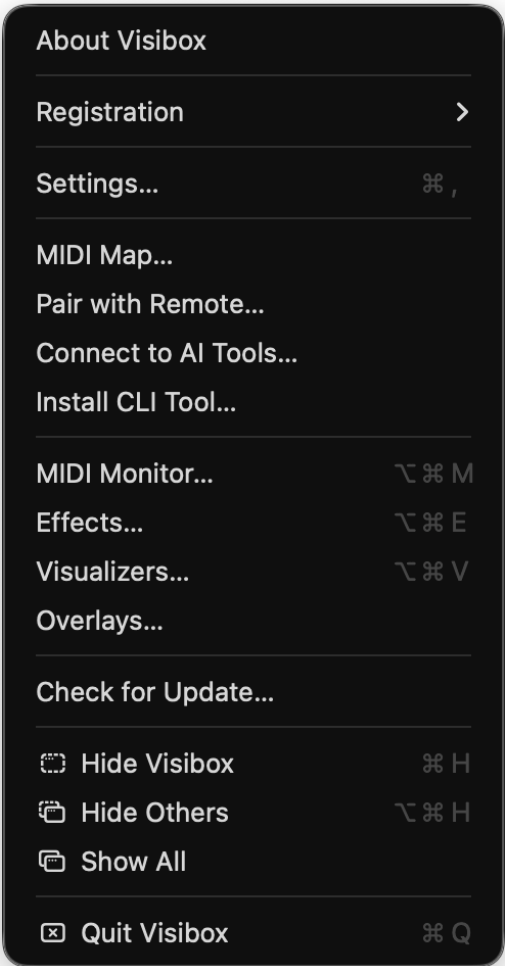
This section covers what every menu option in Visibox does.

Menu wording and layout can change slightly between updates.

Application Menu

NOTE

The Application menu does not exist on Windows. These items are found in the **File menu** and **Help menu** instead.



The Visibox application menu contains functions related to the application itself, such as registration, settings, editor windows, updates, and Quit.

About Visibox

Displays information about the current version of Visibox.

Registration ›

Registration and authorization. Purchase a Pro plan subscription or license here, and manage your account.

See [Your Account](#) for what the submenu holds and what each plan covers.

Settings... ⌘,

Opens the [Settings window](#), the central place to configure audio, devices, MIDI, output, displays, remote access, language, and general preferences. The interface language is set in [General Settings](#); the Language submenu that used to sit in this menu is gone.

MIDI Map...

Opens Settings on the [MIDI](#) category, where you assign notes and CCs to actions. In Visibox 5 this was a window of its own.

Pair with Remote...

Opens Settings on the Remote category, where you pair remote devices and AI tools. See [Remote Pairing](#).

Connect to AI Tools...

Scans for installed AI tools on your machine and registers Visibox's MCP server with each. See [AI Assistants](#) for what you can do after connecting.

Install CLI Tool...

Installs the `visibox` command-line tool for shell scripting. See [Command Line Interface](#).

MIDI Monitor... ⌘⌥M

Opens the [MIDI Monitor](#) window for troubleshooting MIDI connections.

Effects... ⌘⌥E

Opens the [Effects Editor](#) window for browsing, creating, and customizing Effects.

Visualizers... ⌘⌥V

Opens the [Visualizers Editor](#) window for browsing, creating, and customizing Visualizer presets.

Overlays...

Opens the Overlays editor, for the logos and lower thirds that sit above everything the Output shows. It has no keyboard shortcut.

Check for Update...

Checks to see if a newer version of Visibox is available. The same check, plus the version you are running, is in [Updates Settings](#).

Hide Visibox ⌘H

Standard macOS hide command.

Hide Others ⌘⇧H

Hides all other application windows.

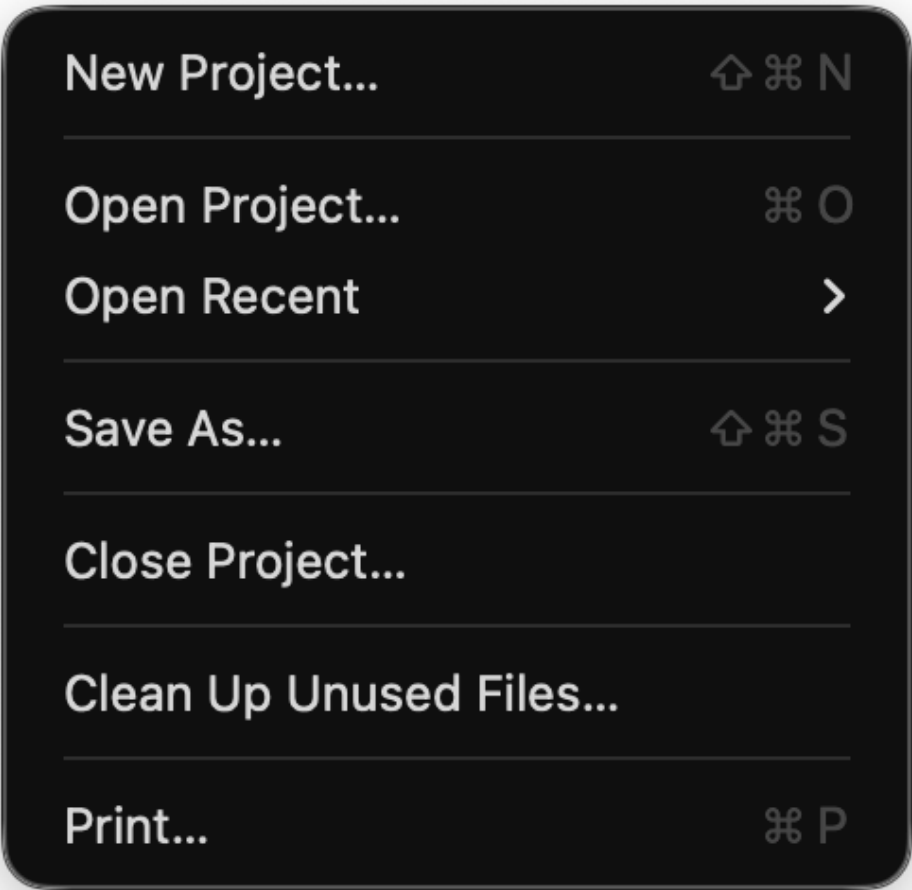
Show All

Shows all hidden application windows.

Quit Visibox ⌘Q

Quits Visibox.

File Menu



NOTE

On Windows, the File menu also includes items that are in the Application menu on Mac, such as Registration, Settings, and editor windows.

New Project... ⇧ ⌘ N

Start a new Project. Visibox asks for a name and a location. The default location is your Documents folder, but you can choose any location on your computer or an external drive.

Open Project... ⌘ O

Open an existing Project. Visibox asks you to select a Project file.

Open Recent ›

Recent Projects are listed here. Choose Clear Menu to clear the list.

Save As... ⌘S

Save the current Project to a new location, media files and all. Use it to make a new version of the current Project, or to move a copy onto another drive.

Close Project...

Closes the current Project.

Clean Up Unused Files...

Clean Up Unused Files... acts like Empty Trash: it deletes media the Project no longer uses, to save space. Visibox keeps removed files until then, so Undo can still bring them back.

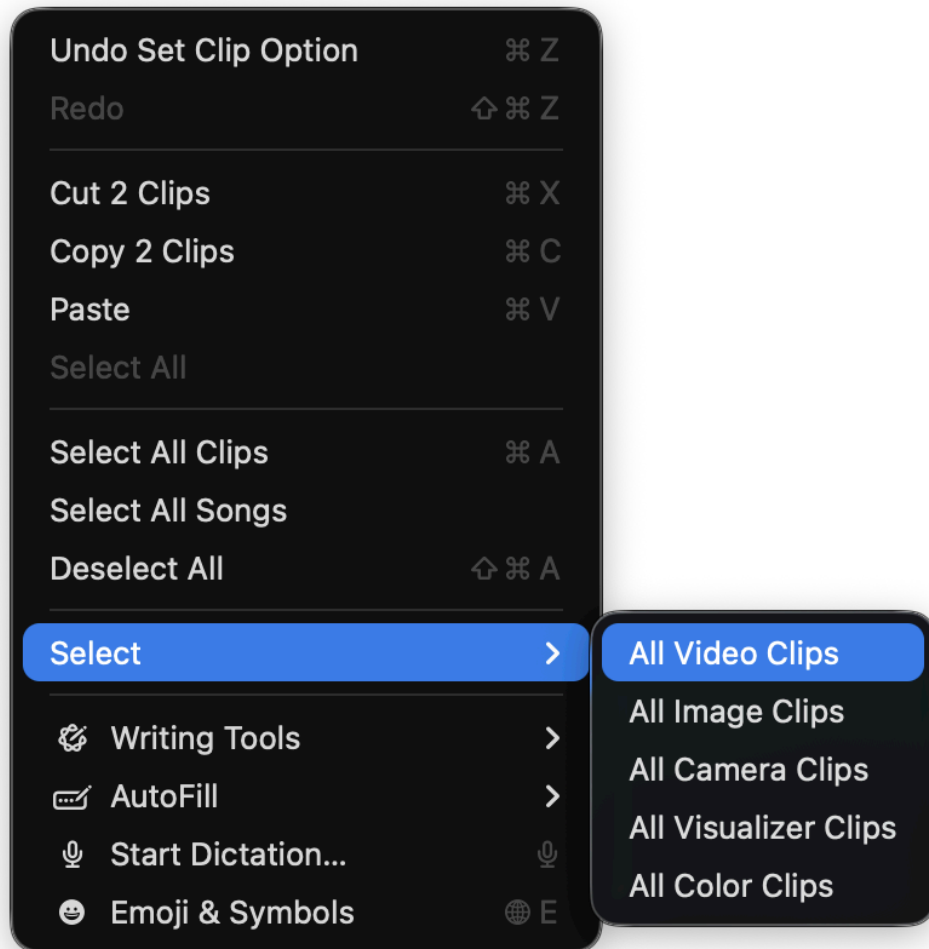
Print... ⌘P

Print a Setlist for your Project: the Project name at the top, the running order in large type, and each Song's Clips beside it as numbered thumbnails. A Song never breaks across a page, and nothing is left off because it was scrolled out of sight in the Controller. Print a copy for everyone in the band, including whoever runs sound.

Windows-only items. On Windows, the File menu also contains these application-level items (on Mac these live in the [Application menu](#)):

- **Registration** › : see [Registration](#).
- **Settings...** , **MIDI Map...** , **Pair with Remote...** , **Connect to AI Tools...** , and **Install CLI Tool...** : no longer a submenu. Settings... sits on its own, then after a separator come MIDI Map..., Pair with Remote..., Connect to AI Tools... and Install CLI Tool.... See [Settings...](#), [MIDI Map...](#), [Pair with Remote...](#), [Connect to AI Tools...](#), and [Install CLI Tool...](#)
- **MIDI Monitor...** : opens the [MIDI Monitor](#) window.
- **Effects...** : opens the [Effects Editor](#).
- **Visualizers...** : opens the [Visualizers Editor](#).
- **Overlays...** : opens the Overlays editor. No keyboard shortcut.
- **Exit** : quits Visibox.

Edit Menu



The Edit menu with the Select submenu expanded.

The Edit menu holds Undo and Redo, the clipboard, and the selection tools for working on many Songs and Clips at once.

Cut, Copy, and Paste do one of two jobs depending on where you are. In a text field (a Song title, the Timeline Editor) they edit text, the way they do in any app. Everywhere else they act on whatever you have selected in the Controller window, and the menu rows rename themselves to say so.

Undo ⌘Z

Reverse the last action. The row names what it will undo, as in “Undo New Song” or “Undo Move Clip”.

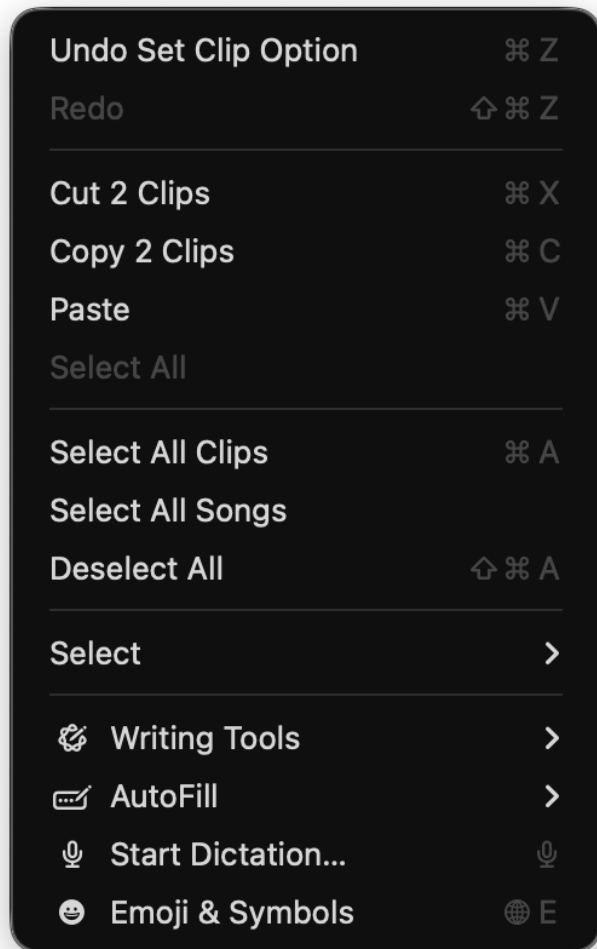
Redo ⇧⌘Z

Put back an action you undid. The row names it.

Cut ⌘X

Remove the selection and put it on the clipboard.

With no text field focused, this cuts your Controller selection, and the row names it: **Cut Clip**, **Cut 3 Clips**, **Cut Song**, **Cut 2 Songs**. With nothing selected it stays a plain, unavailable **Cut**.



With Clips selected, the Cut and Copy rows count what they will act on.

Copy ⌘C

Copy the selection to the clipboard without removing it. It renames itself the same way: **Copy Clip**, **Copy 4 Clips**, **Copy Song**, **Copy 3 Songs**.

Paste ⌘V

Paste the clipboard. The row names what is on it: **Paste Clip**, **Paste 2 Clips**, **Paste Song**, **Paste 3 Songs**.

Clips paste into the active Song. Songs paste into the Project.

Paste also brings in files copied in Finder or File Explorer. When the newest thing on the system clipboard came from another app, Visibox imports it instead of pasting its own Clips.

Moving Songs and Clips between Projects

Copy in one open Project and paste in another. The media files come with what you paste, so nothing is left pointing at the Project you copied from. Large videos take a moment.

Dragging works too: drag a Song by its handle, or a Clip by its grip, from one Project's Controller window into another.

Select All

Select all the text in the text field you are editing. Outside a text field the row is unavailable. To select Clips, use **Select All Clips**.

Select All Clips ⌘A

Select every Clip in the Project.

Select All Songs

Select every Song in the Project.

Deselect All ⌘⇧A

Clear the selection.

Select ›

The Select submenu picks Clips by type, across every Song.

All Video Clips

Select only Video Clips.

All Image Clips

Select only Image Clips.

All Camera Clips

Select only Camera Clips, including HDMI video inputs.

All Visualizer Clips

Select only Visualizer Clips.

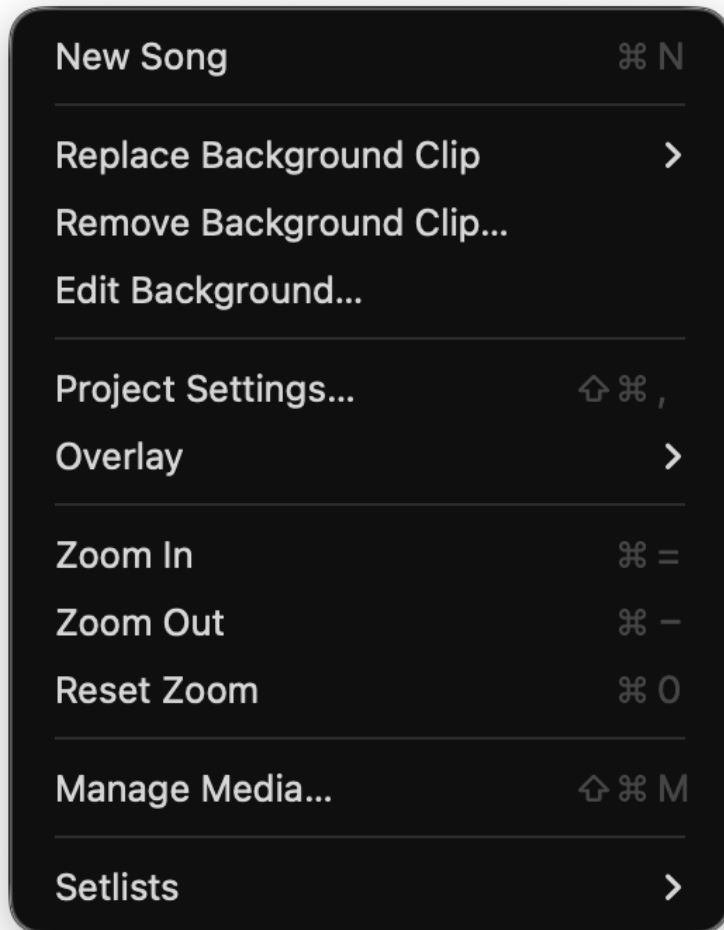
All Color Clips

Select only Color Clips.



Select is built for batch work. To drop every Color Clip from a Project, choose Edit > Select > All Color Clips, then press Delete. To retune a whole set of Clips at once, select them by type and open the Clip Editor Panel.

Project Menu



The Project menu holds the things that belong to the Project as a whole: adding Songs, setting the background, choosing a running order, and opening the windows that edit the Project's own settings.

The controls that decide how the output looks are no longer here. Rotation, flip, enforced aspect ratio, rounded corners and video wall compensation are all in the Settings window now, because they describe the room rather than the Project. See [Output Settings](#) and [Canvas Settings](#).

New Song ⌘N

Adds a new Song to the Project. You will get used to ⌘N, but the menu item is here when that is not an option.

Paste Song

Pastes Songs you have cut or copied into this Project. The row only appears when Visibox is holding Songs on the clipboard, and it names the number it will paste, so a three-Song copy reads “Paste 3 Songs”.

Paste Songs into a different Project from the one you copied them from.

Add Background Clip ›

By default the Output shows black when nothing is playing. A Background Clip fills that space instead.

The Background Clip can be any kind of Clip, not only a video:

Submenu item	What it adds
Image or Video...	Opens a file picker for an image or video file.
Camera	Lists your connected cameras and video inputs. Pick one to make it the background.
Visualizer	Lists your visualizer presets. Pick one to have the background react to audio.
Color	Lists the color presets. Pick one to make it the background.

Once a Background Clip is set, this row reads **Replace Background Clip** and swaps it for a new one.

You can also drag an image or video file from the desktop onto the Output window, or click the Background thumbnail in the Control Bar to add, replace or edit it during a show.

A Background Clip belongs to the whole Project. Songs cannot each have their own.

Remove Background Clip...

Takes the Background Clip out of the Project. The Output shows black again when nothing is playing.

Edit Background...

Opens the Background Clip’s own editor panel: fit mode, rotation, flip, Filters, Effects, trim, speed, and the rest. It is the same set of options any other Clip gets, in the same panel, rather than the nested submenu this used to be.

Project Settings... ⌘,

Opens the [Project Settings](#) window, which holds the Project title, Uppercase Titles, Default Fit Mode and the Control Bus.

Overlay ›

Assigns an Overlay to the whole Project, so it appears over every Song and every Clip that does not override it.

The submenu lists **Inherit**, **None**, and your Overlay library. Overlays stack, so several can be checked at once. See [Overlays](#).

Zoom In ⌘+

Makes the Controller window's content larger.

Zoom Out ⌘-

Makes the Controller window's content smaller.

Reset Zoom ⌘0

Returns the Controller window to 100%.

Manage Media... ↑ ⌘M

Opens the [Media Management](#) window, where you can see every media file in the Project, optimize them, and manage their thumbnails.

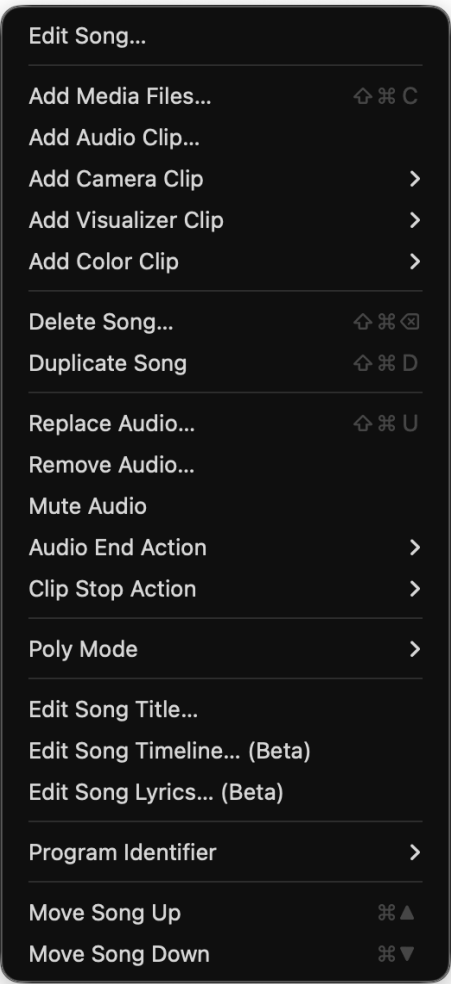
Setlists ›

A Setlist is a saved running order of the Songs in a Project. One Project can hold several, so the same material can be played in a different order at a different venue.

The submenu lists every Setlist in the Project, with the active one marked. Choose one to switch the Controller to that order. Below the list:

- **Save Current Order as New Setlist** saves the running order you are looking at now as a new Setlist.
- **Manage Setlists...** opens the Setlists window, where Setlists are renamed, reordered, and deleted.

Song Menu



The Song menu acts on whichever Songs are selected in the Controller window. Its title follows the selection, so it reads “Song” for one and “Songs” for several.

The menu carries the Song actions, the settings that fit a list of choices, and **Edit Song...**, which opens the [Song Editor Panel](#). The two routes set the same values. What the panel adds is what a menu cannot draw: the audio volume slider, the Overlay picker, and the freeform color picker behind Add Color Clip.

NOTE

New Songs come from **Project > New Song** or N .

Edit Song...

Open the [Song Editor Panel](#) for the selected Songs. Right-clicking a Song does the same thing.

It leads the menu because it is the way to every control the rest of the menu cannot draw.

Adding Clips

These rows add a Clip to the Song. They appear when a single Song is selected.

Add Media Files... C

Pick one or more media files (video or image) to add to the Song. Same result as dragging files from the desktop onto the Song. The picker takes audio files too, and picking a single one asks whether to attach it to the Song or make it a Clip.

Add Audio Clip...

Pick one or more audio files and add each as its own Clip. Unlike a drop, this never asks: it always makes Clips. See [Audio Clips](#).

Add Camera Clip

Lists the cameras and capture devices attached to this computer. Pick one and the Song gets a Clip showing its live picture. See [Cameras](#).

If nothing is connected, the submenu holds a single **No Cameras Found** row.

Add Visualizer Clip

Lists the Visualizers in the Project. Pick one and the Song gets a Clip that draws it live from the audio. See [Visualizers](#).

If the Project has none, the submenu holds a single **No Visualizers Found** row.

Add Color Clip

Lists the named colors. Pick one and the Song gets a solid-color Clip. For a color that is not on the list, use the picker in the [Song Editor Panel](#).

Delete Song...

Delete the selected Songs and every Clip in them. With several selected the row counts them: **Delete 3 Songs....**

Duplicate Song D

Copy the selected Songs, with their Clips and settings. Duplicates point at the same media files as the originals, so a copy costs almost no disk space.

You can also hold  and drag a Song by its handle to duplicate it where you drop it.

Attach Audio...

Attach an audio file to the Song as a backing track that plays while the Song's Clips play. Once a Song has audio, this row reads **Replace Audio....**

Audio files can be:

- Mpeg-3 (mp3)
- Advanced Audio Codec (aac)
- Mpeg-4 audio (m4a)
- WAV (wav)
- FLAC (flac)
- Ogg (oga)
- OPUS (opus)

Mute Audio, Audio End Action and Clip Stop Action are the three rows below. Volume is in the [Song Editor Panel](#), because a menu cannot draw a slider.

Remove Audio...

Take the backing track off the Song. Available only when the Song has one.

Mute Audio

Silence the backing track without removing it. A checkbox, not a list. Available only when the Song has audio.

Audio End Action ›

What happens when the backing track finishes: **Stop Clips**, **Clips Keep Playing**, **Loop Audio**, **Cue Next Song**, or **Start Next Song**. Available only when the Song has audio. See [Audio End Action](#).

Clip Stop Action ›

What happens to the backing track when the last playing Clip stops: **Stop Audio** or **Audio Keeps Playing**. Available only when the Song has audio. See [Clip Stop Action](#).

Poly Mode ›

How this Song puts more than one Clip on screen at once: **Mono**, the default, plus **Blend**, **Mosaic**, **Horizontal Stripes**, **Vertical Stripes**, **Stack**, and **Spots**. See [Poly Mode](#).

Edit Song Title...

Put the Song title into edit mode. Double-clicking the title in the Controller window does the same. Available for a single Song.

Edit Song Timeline... (Beta)

Open the Timeline Editor for the Song, where you write a timed cue list that fires triggers, playback changes, and MIDI as the Song runs. Available for a single Song.

Turn on Settings > General > Enable Beta Features to see this.

Edit Song Lyrics... (Beta)

Open the same window on its Lyrics tab, where you write the Song's chords and lyrics. Available for a single Song.

Turn on Settings > General > Enable Beta Features to see this.

Program Identifier ›

Give the Song a MIDI program number, so a program change message selects it. The list runs from None through 1 to 128, or 0 to 127 with **Program Change numbers start at 0** turned on in [Settings > MIDI](#). A number another Song already holds is listed beside that Song's title, anywhere in the Project: a Song outside the active Setlist reads "(not in setlist)". Picking a number that is taken swaps the two Songs' numbers rather than creating a duplicate. Available for a single Song.

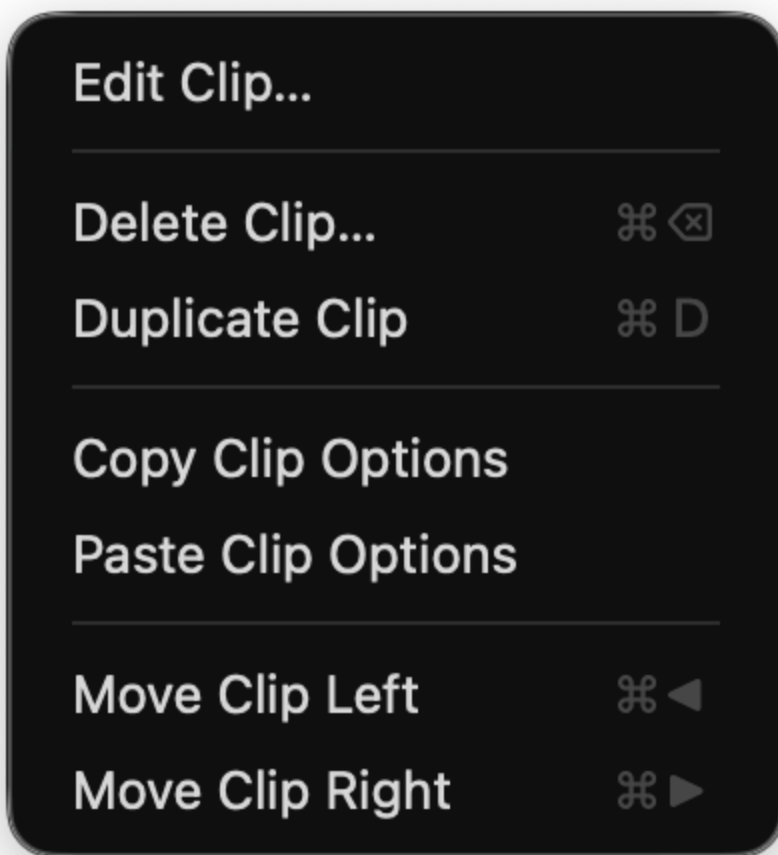
See [Selecting Songs via MIDI Program Change](#).

Move Song Up ⌘↑

Move Song Down ⌘↓

Move the selected Songs through the running order. Dragging a Song frame by its handle does the same. The running order belongs to the active Setlist. See [Setlists](#).

Clip Menu



The Clip menu acts on the Clips you have selected in the Controller. It carries the Clip actions and **Edit Clip...**, which opens the [Clip Editor Panel](#) where every Clip setting lives.

The playback options themselves (End Action, Retrigger Action, Resume Action, Direction, Fit Mode, Mute, and the rest) are not in this menu. They are set in the Clip Editor Panel, and each one is explained on [Clip Playback Options](#). Right-clicking a Clip thumbnail opens the panel, and that is the faster route during a show.

When no Project is open, the menu shows a single **No Clips Selected** row.

Edit Clip...

Opens the [Clip Editor Panel](#) for the selection. This is where you set playback, trim, audio, display, Filters, Effects, Overlay, and Transitions, along with [Launch Mode](#).

It leads the menu because it is the way to every control the rest of the menu cannot draw. See [Clip Playback Options](#) for what each setting does.

Delete Clip... ⌘⌫

Deletes the selected Clips. The label counts them, so it reads **Delete 3 Clips...** with three selected.

Duplicate Clip ⌘D

Copies the selected Clips into the same Song. All copies share one media file on disk, so a duplicate costs no extra space.

Copy Clip Options

Captures the selected Clip's settings, without its media. Available with a single Clip selected.

Paste Clip Options

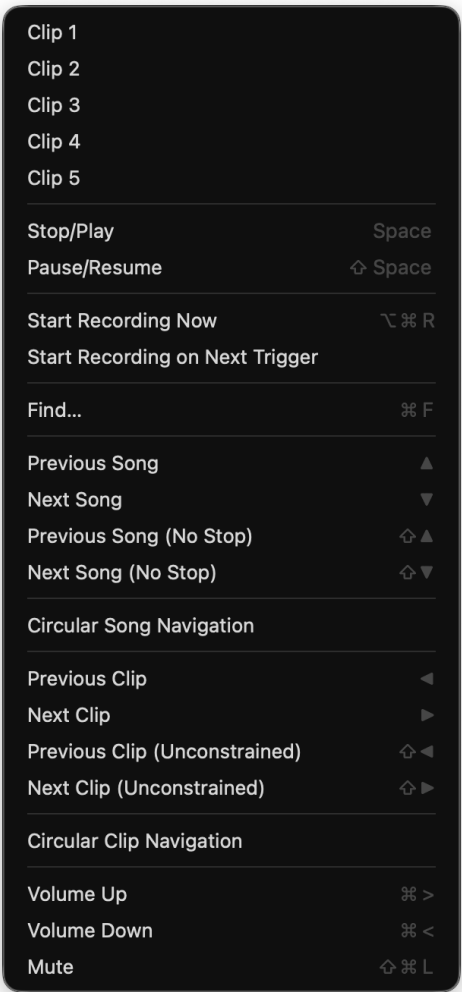
Applies the captured settings to every Clip in the current selection. This is the quick way to make a whole Song behave alike once one Clip is right.

Move Clip Left ⌘←

Move Clip Right ⌘→

Moves the selected Clips within their Song. Dragging a Clip thumbnail does the same thing, though the Clip tiles play on press, so drag from the grip that appears when you hover a tile.

Play Menu



The Play menu is the menu-bar route to the Clip and Song playback functions you will mostly reach with the keyboard or a MIDI controller.

Clip 1 through 5

Clips are triggered with the number keys, or by MIDI. The 1 key triggers the first Clip, 2 the second, and so on. Keys 6 through 9 trigger their Clips too, though those rows are hidden in the menu. These items are here as an alternative when the keys are not available.

A Clip whose [Launch Mode](#) is Gate plays only while its number key is held, and stops when you let go.

Stop/Play Space

Pause/Resume ↑Space

With nothing playing, the space bar starts the first Clip of the current Song. With a Clip playing, it stops it. That makes the space bar a quick way to say stop without jumping to another Song.  pauses the current Clip and resumes it.



TIP

The space bar is handy when you are using Visibox mainly for backing tracks. Most Songs then have a single Clip, and the space bar gets the track going or stops it if something goes wrong.

Recording

These rows record the Output to a file. Which ones you see depends on what recording is doing right now. The main row of each state is on  ⌘R, so the one shortcut starts a recording and stops it again. Start Recording on Next Trigger has no shortcut of its own.

Start Recording Now ⌘R

Starts recording the Output immediately.

Start Recording on Next Trigger

Arms the recorder. Nothing is written until the next Clip is triggered, so the file starts on your first cue rather than on the click that set it up.

Cancel Armed Recording ⌘R

Disarms a recorder that is waiting for a trigger. Appears in place of the two rows above once recording is armed.

Stop Recording ⌘R

Ends the recording and writes the file. Appears while recording is running.

The Controller's control bar has a [Record button](#) that does the same thing.

Find... ⌘F

Opens the Find box in the Control Bar so you can search for and jump to any Song or Clip in the Project. See [Find](#).

Previous Song ↑

Next Song ↓

Previous Song (No Stop) ↑↑

Next Song (No Stop) ↑↓

These move between Songs, though the arrow keys are usually easier. Previous Song (↑) and Next Song (↓) cue the neighbouring Song and stop whatever Clip is playing. Hold shift to cue the next Song without stopping the current Clip.

You can also click a Song to select it. Shift-clicking selects it without stopping anything.

Circular Song Navigation

With this on, Next Song from the last Song wraps around to the first, and Previous Song from the first Song wraps to the last.

Previous Clip ←

Next Clip →

Previous Clip (Unconstrained) ↑←

Next Clip (Unconstrained) ↑→

The number keys and MIDI give you any Clip in the current Song at any time. These items step through them instead: Previous Clip (←) and Next Clip (→) stay inside the current Song. Hold shift to step past its edges into the neighbouring Songs.

Circular Clip Navigation

With this on, Next Clip from the last Clip wraps around to the first, and Previous Clip from the first wraps to the last. It also applies to [End Action](#), so a last Clip set to Start Next Clip plays the Song's first Clip when it finishes.

Volume Up ↑⌘>

Raises the master output level a step.

Volume Down ↑⌘<

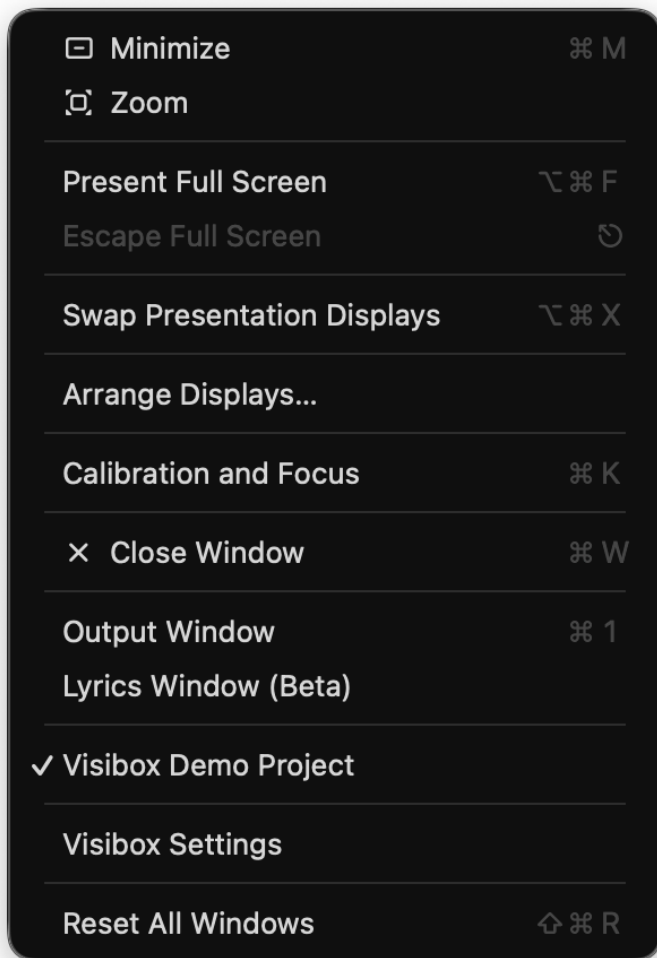
Lowers the master output level a step.

Mute ↑ ⌘ L

Mutes and unmutes all audio output. The volume slider keeps its position while muted, and moving it unmutes.

These three rows sit after a separator at the end of the menu. In Visibox 5 they were in the Application menu's Settings submenu.

Window Menu



The Window menu handles presentation, display arrangement, and getting to any window Visibox has open.

Minimize ⌘M

Minimizes the front window. This is the operating system's own command, so it behaves the way it does in any other app.

Zoom

Toggles the front window between its current size and the largest size that fits the display. This is the macOS command that Windows calls Maximize. It is a Mac-only row: on Windows, use the Maximize button in the window's title bar.

Present Full Screen ⌘F

Escape Full Screen

Select Present Full Screen (⌘F) to present your Output and Controller windows full screen on different displays. If only one display is connected, only the Output goes full screen.

Once you are in full screen, Escape Full Screen brings you back out. The  key does the same, except while you are editing text.

Swap Presentation Displays ⌘X

Swap Presentation Displays (⌘X) trades the Controller and Output displays when they land on the wrong screens.

Arrange Displays...

Opens the Canvas tab of the Settings window, where a map of your connected displays shows how they are laid out and which one has which job. Assigning the Output display, assigning the Controller display, and flashing an identifying number onto each screen all happen there.

Arrange Displays... replaces the Output Display, Controller Display, and Identify Displays items from earlier versions. It stays available with a single display, since it is also where you check what Visibox is about to do before you plug the projector in.

Calibration and Focus ⌘K

In a typical performance you may have only minutes to set up, focus, and calibrate a projector. Calibration and Focus (⌘K) projects a test image: fine lines for focus, bright colors for color calibration, and grayscale steps for contrast. Aspect ratio markings help you confirm the shape you are actually filling. The row is a checkbox: choose it again to take the test image away.

Output Window Resize

Sets the Output window to a standard pixel size, from 640 x 480 up to whatever your display can hold. Sizes too large for the current display are not listed.

This row, **Output Window Aspect Ratio**, and **Show Output Window Controls** appear only while the Output window is showing. Calibration and Focus is always there, because it can bring the window up itself.

Output Window Aspect Ratio

Constrains the Output window to 1:1, 4:3, 16:9, or 16:10 as you resize it. Unconstrained lets you drag it to any shape.

Show Output Window Controls

Shows or hides the Output window's own title bar and controls. Uncheck it to hide output controls once the Output is on a projector.

Close Window

Closes the front window.

Output Window

Brings the Output window to the front.

Lyrics Window (Beta)

Opens the Lyrics window, which shows the current Song's lyrics and chords. Turn on Settings > General > Enable Beta Features to see this.

Open windows

Below those, the menu lists everything Visibox currently has open: the Controller window for each open Project, then tool windows such as Settings and the MIDI Monitor, then the editors.

- A **checkmark** marks the front window.
- A **diamond** (◊) marks a window that is minimized.

Click any name to bring that window forward, restoring it first if it was minimized.

Editor tabs are listed one by one

Effects, Visualizers, and Overlays share one window, and that window is named after whichever tab is in front. Listing it as a window would hide the other two editors behind the name of the third, so the menu lists **one row per open tab** instead. Click any of them and Visibox opens the window, brings that tab to the front, and focuses it. The checkmark appears on the tab that is both active in its window and in front on screen. See [The Editor Window](#).

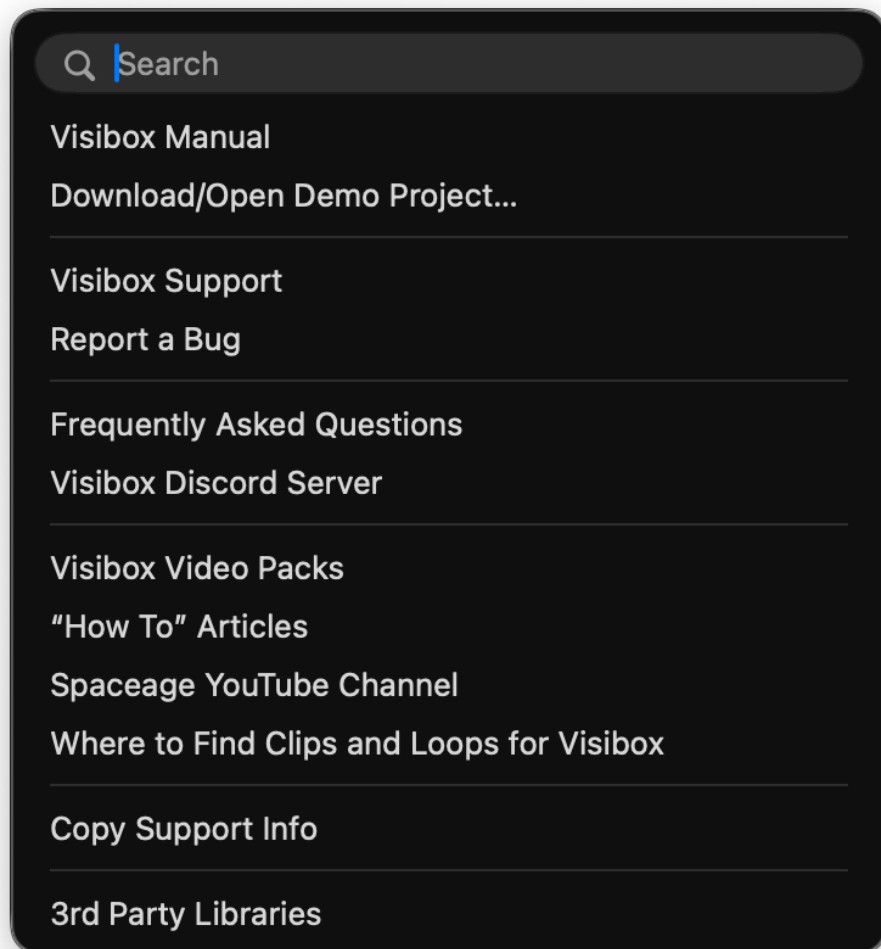
Reset All Windows

A panic button for Visibox. Use **Reset All Windows** to reload every window. If something is not behaving, this will usually fix it.

Window bounds persistence

Visibox remembers the position, size, and stacking order of its windows across restarts, so reopening the app puts everything back where you left it.

Help Menu



Visibox Manual

You are currently reading the manual! Good work.

Download/Open Demo Project...

Downloads and opens a demo Project so you can try Visibox before you bring in your own media.

Visibox Support

Opens the built-in [Support form](#), where you send us a question or request. Visibox attaches your system information automatically, to help with troubleshooting.

Report a Bug

Opens the built-in [Bug Report form](#) with structured fields for describing the issue, steps to reproduce, and expected versus actual behavior. Visibox attaches your system information automatically.

Frequently Asked Questions

The [Visibox FAQ](#) has answers to common questions.

Visibox Discord Server

Join the [Visibox Discord server](#) to chat with other users and the Spaceage team.

Visibox Video Packs

Free [video loops and content packs](#) to get you started.

“How To” Articles

[These articles](#) have tips and tricks for using Visibox.

Spaceage YouTube Channel

Tutorials and videos on the [Spaceage YouTube channel](#).

Where to Find Clips and Loops for Visibox

Need content for your Project? See this list of [content providers](#).

Copy Support Info

Copies detailed information about your system and Visibox setup to the clipboard. Use it to contact support outside the built-in forms. See [Support & Bug Reports](#) for more.

3rd Party Libraries

A list of the third-party open-source libraries distributed in Visibox.

About Visibox displays information about the current version of Visibox. On Mac, this item is in the [Application menu](#).