User Manual

ATTACCA 1.0.0 — 2026-09-09

Illustrations: the screenshots for this edition are being captured and will be added in a future revision. Every place one belongs is marked in the text; the instructions themselves are complete and verified against the application.

Contents

Getting Started

What is ATTACCA?

ATTACCA is a Windows-native show control application. It plays audio, video, images, and text; controls lighting rigs over Art-Net, sACN, and USB-DMX; and speaks the languages of a modern show network — OSC, MIDI, MIDI Show Control, timecode (MTC/LTC), NDI, serial, HTTP, and raw network messages. Everything is organized into a list of cues, and everything fires from one **GO** button.

You build your show once in Edit Mode — cue by cue, with waveforms, fade curves, geometry, and triggers — then lock it down in Show Mode and run the whole performance from `Space`.

[Illustration to be added in a future revision]

System requirements

Requirement	Details
Operating system	Windows 10 or Windows 11, 64-bit
Runtime	None to install — ATTACCA is fully self-contained (no .NET download, no codec packs)
Memory	8 GB RAM recommended
Audio	Any Windows audio output device. You pick the output device inside ATTACCA (**View → Audio Output Device**)
Video	A DirectX 11-capable GPU. Video decoding is done in software, so a faster CPU helps with high-resolution video. A second display for video output is strongly recommended
Lighting / network	A wired network interface is recommended for Art-Net and sACN output. Enttec DMX USB Pro interfaces are supported over USB
MIDI	Standard Windows MIDI devices (USB interfaces, virtual ports such as loopMIDI)

ATTACCA bundles everything it needs, including its own media decoders. You do not need to install codec packs, and installing one will not change which files ATTACCA can play.

Installing ATTACCA

ATTACCA ships in two forms. Both are the full application — same features, same license.

	Installer (`ATTACCA.msi`)	Portable (`ATTACCA.exe`)
Download size	~126 MB	~158 MB
Admin rights	Required (installs for all users)	Not required

	Installer (`ATTACCA.msi`)	Portable (`ATTACCA.exe`)
`.atca` double-click opens ATTACCA	Yes	No
Start Menu / Desktop shortcuts	Yes	No
Best for	Your main show machine	USB sticks, locked-down or borrowed machines

A note on SmartScreen and antivirus

ATTACCA is digitally signed by its developer (Cody Yeager). Because it is a newer application without a long download history, Windows SmartScreen may still show a "Windows protected your PC" screen the first time you run the installer or the portable exe. Click **More info**, verify the publisher, then click **Run anyway**. Some antivirus products will also scan the file on first launch — this is normal reputation-building behavior, not a sign of a problem.

Installing with the MSI

1. Download `ATTACCA.msi` and double-click it.
2. If SmartScreen appears, click **More info** → **Run anyway**.
3. Accept the license agreement.
4. Choose the install folder (default: `C:\Program Files\ATTACCA`) and click through to **Install**. Windows will ask for administrator approval.
5. On the final screen, leave **Launch ATTACCA** checked and click **Finish**.

> *[Illustration to be added in a future revision]*

The installer creates a Start Menu shortcut and a Desktop shortcut, and registers the `.atca` workspace file type — double-clicking any workspace file opens it in ATTACCA.

Updating: run a newer `ATTACCA.msi` over your existing install. It upgrades in place — no uninstall needed, and your workspaces, preferences, and license are untouched. Installing an older version over a newer one is blocked.

Uninstalling: remove ATTACCA from Windows **Settings** → **Apps** as usual. Uninstalling removes the program only — your workspaces, preferences, shortcuts, license, and logs in `%APPDATA%\ATTACCA` are preserved, so reinstalling later picks up exactly where you left off.

Using the portable exe

The portable build is a single `ATTACCA.exe` — copy it anywhere (including a USB stick) and run it. No installation, no admin rights.

Two things to know about the first launch:

- The exe self-extracts its runtime to `%TEMP%\.net\` the first time it runs on a machine. This causes a one-time delay of several seconds; later launches are fast.
- Antivirus software may scan the extracted files during that first launch, extending the delay. This is a heads-up, not a fault.

Because the portable build doesn't register anything with Windows, double-clicking `.atca` files won't open ATTACCA — open workspaces from inside the app instead.

First launch: the Welcome window

Whenever you start ATTACCA without opening a specific workspace file, the Welcome window appears.

[Illustration to be added in a future revision]

It contains:

- **New Workspace** — create a fresh, empty show.
- **Open Workspace** — browse for an existing `.atca` file.
- **Recent Workspaces** — double-click any entry to open it. Files that have been moved or deleted are dimmed in red; before your first show it reads *No recent workspaces*.
- The version number, in the footer.

Closing the Welcome window without choosing anything exits ATTACCA. Note that the recent-files list lives here — there is no Recent Files entry in the main window's File menu.

Licensing and tiers

The three tiers

ATTACCA has three tiers. Each is a one-time purchase that never expires.

Tier	Price	What it adds
Free	$0	A complete audio show-control rig: Audio cues, all the structural and transport cue types, per-cue audio effects, unlimited cue lists and carts, triggers, Show Mode, the Show Timer
Standard	$199 one-time	The video family — Video, Camera, Image, and Text cues — plus audio markers and the Web Remote
Pro	$399 one-time	Lighting (Art-Net/sACN/USB-DMX), MIDI, OSC control, timecode, Lua scripting, Serial and HTTP cues, collaboration, and broadcast output
Standard → Pro upgrade	$200	Your existing key becomes Pro — no new key is issued
Update plan	$49/year	Optional — only needed for **new** versions. The version you own keeps working forever

There is no monthly subscription. When your update plan lapses, nothing turns off — you simply stay on the version you have.

The full feature-by-feature table is in *Licensing & Activation*, along with machine management, offline behavior, and the update plan.

Cue types above your tier are never hidden: they appear dimmed with a padlock glyph, and the padlock's tooltip names the tier that unlocks them — *This cue requires ATTACCA Standard* or *This cue requires ATTACCA Pro*. Clicking such a cue's toolbar button opens a dialog explaining the feature, with **Enter License Key**, **Upgrade…**, and **Close** buttons.

[Illustration to be added in a future revision]

> **Why it works this way:** locked cues in a loaded workspace stay visible (dimmed, with the lock) rather than disappearing — so opening a Pro workspace on a Free machine never hides or deletes anything.

Beta testers: the beta program ended on September 1, 2026 — beta keys no longer activate, and a machine that still has one runs as Free until a purchased license is activated.

Activating a license

1. Open **View** → **License…**.
2. Paste your key into the box and click **Activate**.
3. The window shows your tier, the licensed email, the purchase date, and how long updates are included. The new tier takes effect immediately — no restart.

[Illustration to be added in a future revision]

License keys are long — this is normal; they are cryptographically signed to prevent tampering. Activation needs an internet connection **once**; after that ATTACCA runs fully licensed offline forever. Your license is stored in `%APPDATA%\ATTACCA` and survives uninstall/reinstall.

Until a key is entered, a dismissible banner reminds you that you are running in Free mode.

For machine seats, moving a license to a new computer, upgrades, and the update plan, see *Licensing & Activation*.

Your first show in 10 minutes

This walkthrough builds a two-cue show: a music track, then a video on a second display. You'll need an audio file, a video file, and (for the video steps) a Standard, Pro, or beta license — Free-tier users can simply stop after step 6.

1. Launch ATTACCA and click **New Workspace** on the Welcome window. You get an empty cue list in Edit Mode.
2. Click the **Audio** button in the cue toolbar (or press `Ctrl+1`). An empty Audio cue appears in the list, numbered 1, and is selected.

[Illustration to be added in a future revision]

3. Look below the cue list: the inspector shows the selected cue's tabs. Click the **I/O** tab, then click **Browse…** and pick your audio file. The cue's name, duration, and waveform fill in automatically.

[Illustration to be added in a future revision]

4. Press `Space` (or click the big green **GO** button). Your track plays, the row shows a ▶ icon, and the **Duration** column counts down live.
5. Press `Esc` to stop it. `Esc` is Panic — it fade-stops everything (half a second by default). You'll use it constantly in rehearsal.
6. Click the **Video** button in the toolbar (or press `Ctrl+3`) to add a Video cue. *(Free tier: this opens the upgrade dialog instead — skip to step 9.)*
7. With the Video cue selected, open its **I/O** tab, click **Browse…**, and pick a video file. A first-frame preview appears.
8. Still on the **I/O** tab, use the **Display:** dropdown to choose which monitor the video plays on (entries read "Display 1 — …", "Display 2 — …").

[Illustration to be added in a future revision]

9. Click the first cue's row. Clicking a cue moves the **playhead** — the highlighted row that GO will fire next. The standby readout at the top of the window confirms what's next.
10. Press `Space`. The music starts and the playhead advances to the video cue. Press `Space` again — the video opens fullscreen on its assigned display, over the music.

[Illustration to be added in a future revision]

11. Press `Esc` to bring everything down, then press `Ctrl+S` and save your workspace as `MyFirstShow.atca`.

That's the whole model: build cues, aim the playhead, press GO. Everything else in this manual is refinement.

Common issues

- **"Windows protected your PC" when launching the installer** — SmartScreen reputation notice. Click **More info**, confirm the publisher, then **Run anyway**.
- **The portable exe takes ages to start the first time** — it's extracting its runtime to `%TEMP%\.net\` and your antivirus is scanning it. One-time only.
- **Double-clicking a `.atca` file does nothing** — you're using the portable exe, which doesn't register the file type. Use the MSI install, or open workspaces from inside ATTACCA.
- **Some toolbar buttons open an upgrade dialog; some cues are dimmed with a lock** — those cue types need a higher tier than the one you're running. Hover the padlock to see which tier, then enter a license key via **View → License…**.
- **Audio comes out of the wrong output** — pick the right device under **View → Audio Output Device**. The change is immediate.
- **Closing the Welcome window quits the app** — that's by design; choose **New Workspace** or **Open Workspace** instead.
- **My license key looks absurdly long** — normal. Keys are cryptographically signed.

The Interface

The main window at a glance

ATTACCA is a single dark window built around one idea: the cue list in the middle, the **GO** button at the top, and everything else arranged around them.

> *[Illustration to be added in a future revision]*

From top to bottom:

1. **Title bar** — shows the app name, version, workspace file name, and current mode, e.g. `ATTACCA 1.0.0 – MyShow [Edit]`. The mode token flips to `[Show]` in Show Mode. There is no asterisk for unsaved changes — ATTACCA prompts you instead when closing with changes pending.
2. **Menu bar** — File, Edit, View.
3. **Transport** — GO, pause/resume, stop, Panic, reset, plus the standby readout showing which cue fires next.
4. **Cue toolbar** — one button per cue type.
5. **Cue list** — the heart of the window.
6. **Inspector** — properties of the selected cue, docked below the list (Edit Mode only).
7. **Sidebar** — cue lists/carts, active cues, and an overview, on the right.
8. **Status bar** — mode toggles, workspace stats, Web Remote and collaboration indicators.

The menus

File

Item	Shortcut
New	`Ctrl+N`
Open…	`Ctrl+O`
Save	`Ctrl+S`
Save As…	`Ctrl+Shift+S`
Export Show Report…	`Ctrl+Shift+E`
Check for Updates…	
Exit	`Alt+F4`

New, Open, Save, and Save As are available in Edit Mode only. Note there is no Recent Files submenu — recent workspaces live on the Welcome window.

Edit (the whole menu is disabled in Show Mode)

Item	Shortcut
Undo / **Redo**	`Ctrl+Z` / `Ctrl+Y`
Cut / **Copy** / **Paste**	`Ctrl+X` / `Ctrl+C` / `Ctrl+V`
Duplicate	`Ctrl+D`
Delete	`Del`
Preferences…	`Ctrl+,`

View

Item	Notes
Toggle Inspector (`Ctrl+I`)	Edit Mode only
Show Timer (`Ctrl+T`)	Opens/closes the Show Timer window
Edit Mode	Toggles Edit/Show Mode
Audio Output Device ▶	Submenu of detected audio devices — switching is immediate
Audio Settings…, **OSC Settings…**, **MIDI Settings…**, **Timecode Settings…**, **Lighting Settings…**, **Broadcast Settings…**	Each jumps to the matching Preferences tab
Light Dashboard…, **Light Patch…**	Pro features
Fixture Library…, **DMX Status…**	
Collaboration…	
About ATTACCA…	Version, credits, and license activation

There is deliberately **no Cue menu** — cues are created from the cue toolbar or keyboard shortcuts (see *Working with cues*).

Transport and GO

[Illustration to be added in a future revision]

- **GO** — fires the cue at the playhead (`Space`). The button turns red while cues are running.
- ⏸ / ▶ — **Pause All** (`[`) / **Resume All** (`]`).
- ■ — **Stop Selected**.
- ⚠ — **Panic — Stop All** (`Esc`): fade-stops everything. Press `Esc` twice within a second for an instant hard stop.
- ⏮ — **Reset All**: stop everything and return the playhead to the top.
- **Standby readout** — the number and name of the cue GO will fire next, or *End of list*. Below it, the playhead cue's notes (or *No notes*) — handy for operator warnings you type into a cue's Notes field.

A second, larger set of the same transport buttons sits at the bottom of the sidebar for touch-friendly operation.

The cue toolbar

One text button per cue type — 31 in all — grouped left to right: audio (**Group**, **Audio**, **Mic**), video (**Video**, **Camera**, **Image**, **Text**), lighting (**Light**, **Fade**, **Effect**), communication (**Network**, **MIDI**, **MIDI File**, **Timecode**), transport-control cues (**Start**, **Stop**, **Pause**, **Load**, **Reset**, **Devamp**, **GoTo**, **Target**, **Arm**, **Disarm**), and utility (**Wait**, **Memo**, **Script**, **Serial**, **HTTP**, **OSC**, **Timer**). Hover any button for its tooltip and shortcut; eight commonly used types have default shortcuts on `Ctrl+0` – `Ctrl+9` (`Ctrl+2` and `Ctrl+8` are intentionally unassigned — see below).

Four buttons — **Mic**, **Network**, **Devamp**, and **Target** — are visible but disabled: those cue types are not available in this version. Existing cues of these types in older workspaces still load normally.

The cue list

[Illustration to be added in a future revision]

Columns, left to right:

Column	Shows
(color bar)	A thin strip in the cue's assigned color; the whole row is also gently tinted
(status)	▶ while the cue runs, ⏸ while paused, or a padlock glyph on cues above your license tier (*This cue requires ATTACCA Standard / …Pro*)
Type	The cue type
#	The cue number — free text, so `2.5`, `A1`, or `Intermission` are all valid
Name	The cue's name, indented inside groups; group rows get a ▶/▼ arrow to collapse/expand their children
Target	The file or cue this cue acts on
Pre-Wait	Delay before the action starts — counts down live while running
Duration	The action's length — counts down live, in green
Post-Wait	Delay before auto-continue — counts down live
Marker	Live countdown (amber) to the next audio marker in a playing audio cue
▶	The continue-mode glyph (yellow) for cues that auto-continue or auto-follow

Row cues you can read at a glance: a green left edge = running, yellow = paused, a highlighted bold row = the playhead, faded rows = disarmed, and dimmed rows with a padlock = above your license tier.

Clicking a row selects it *and* moves the playhead; right-click → **Go To Cue** does the same from the context menu. Only **GO**, hotkeys, and triggers actually fire cues — you can click around a list safely mid-show.

The sidebar

Three tabs on the right side of the window:

- **Cue Lists** — every cue list and cue cart in the workspace (carts are tagged *(Cart)*). Double-click a name to rename it. Buttons below: **+ List**, **+ Cart**, **Delete**. Right-click an item for **Rename** / **Delete**. Deleting asks for confirmation — it removes the list and all its cues.
- **Active Cues** — everything currently playing: number, name, state, elapsed/remaining time, a draggable progress bar to seek, and per-cue ⏸/▶ and ■ buttons.
- **Cue List** — a compact overview (#, Name, Type, Duration) of the current list.

> *[Illustration to be added in a future revision]*

When you select a cue **cart** in the sidebar, the main area switches from a list to a clickable grid — the header reads e.g. `Walk-in music — 4×4 grid`. Clicking a filled cell fires that cue instantly; empty cells show `+` (clicking one in Edit Mode creates an Audio cue there).

> *[Illustration to be added in a future revision]*

The inspector

The inspector docks below the cue list and shows the selected cue's properties in tabs. Every cue type has **Basics** (number, name, timing, color, notes) and **Triggers**; media and lighting cues add their own tabs — an Audio cue shows **I/O**, **Time & Loops**, **Levels**, and **Effects**; a Video cue shows **I/O**, **Audio**, **Geometry**, **Time & Loops**, and **Effects**; and so on. The header shows `3 • Intro Music` for a single cue or `4 cues selected` for a multi-selection (which edits Basics/Triggers across all of them at once).

Toggle it with `Ctrl+I` or **View → Toggle Inspector**, and drag the splitter above it to resize. The inspector is an Edit Mode tool — it's disabled in Show Mode.

> *[Illustration to be added in a future revision]*

The status bar

Five regions across the bottom:

1. **Edit** / **Show** toggle buttons — the quickest way to switch modes.
2. Workspace stats, e.g. `12 cues in 2 lists and 1 cart`.
3. The Web Remote URL when the Web Remote is enabled, e.g. `http://192.168.1.20:8080` (marked `(local only)` when restricted to this machine).
4. The collaboration role — `PRIMARY` or `REMOTE` — when a collaboration session is active.
5. A red **SHOW MODE** badge while in Show Mode.

Notification bars

Two bars can appear under the toolbar: the Free-tier license banner (*Running in Free mode…*, with **Dismiss**) and the update bar (*ATTACCA 1.1.0 is available.* with **Download** / **Dismiss**) when a new release is found.

Workspaces

A **workspace** is your whole show in one `.atca` file: every cue list, cart, cue, trigger, and most settings.

- **Create** one from the Welcome window (**New Workspace**) or **File → New** (`Ctrl+N`).
- **Save** with `Ctrl+S` ; the first save asks for a location, like any Windows app. **Save As…** (`Ctrl+Shift+S`) makes a copy under a new name.
- **Open** with `Ctrl+O` , by double-clicking a `.atca` file (MSI installs), or from the Welcome window's recent list.

Saving is **atomic**: ATTACCA writes to a temporary file and swaps it in only once the write succeeds. A crash or power cut mid-save can never leave you with a half-written, corrupted show file.

> **Why it works this way:** the `.atca` format is plain JSON text (on a single line). It's just a file — copy it, version it, email it, back it up like anything else. Keep it in a folder alongside your media and the whole folder becomes your portable show (see *Media paths* below).

If you open a workspace that's already open in another ATTACCA instance, you'll be warned — running two copies against the same file risks one overwriting the other's saves.

Autosave, backups, and crash recovery

Three overlapping safety nets protect your show:

Autosave. ATTACCA periodically writes a `.autosave` companion file next to your workspace (every minute by default; change it under **Preferences → General → Autosave**, from 30 seconds to 10 minutes, or off). Your real file is only written when *you* save — autosave protects the gap since your last `Ctrl+S` .

Crash recovery. If ATTACCA didn't shut down cleanly, the next time you open that workspace you'll be asked: *A more recent autosave was found for this workspace… Would you like to recover from the autosave?* Choose **Yes** to load the autosaved state (then save it properly), or **No** to load your last explicit save.

[Illustration to be added in a future revision]

Backups. Every time you save over an existing workspace, ATTACCA first snapshots the previous version. The 10 most recent timestamped backups are kept per workspace, in `%APPDATA%\ATTACCA\Backups` . If a save ever goes wrong at the worst moment, yesterday's show is still there.

Safe Mode. If ATTACCA crashes repeatedly (three times within an hour), the next launch silently enters Safe Mode: the app starts normally, but the video, camera, and effects engines and the Web Remote stay offline so you can open your workspace, save it, and sort out the trouble. There's no dialog announcing it — if video cues suddenly won't fire right after a string of crashes, Safe Mode is why; restart the app once things are stable.

Media paths and the Relink Media dialog

Cues store where their media files are, and every save also records each file's location *relative to the workspace file*. That gives you the golden rule for portable shows:

Keep your media in the show folder. Put `MyShow.atca` and your audio/video files in one folder (subfolders are fine). Copy that folder to another drive or machine and everything just works — ATTACCA finds the files by their relative position even though the absolute paths changed.

When files genuinely can't be found (moved, renamed, on an unplugged drive), affected cues are flagged broken, and the **Relink Media** dialog opens automatically after the workspace loads.

> *[Illustration to be added in a future revision]*

Each broken cue appears as a card:

- Line 1: the cue, formatted `Cue 5 — "Intro Music" (Audio)`.
- Line 2: the old (missing) path. Line 3: the proposed new path once a match is found.
- A per-row status: *Missing*, *Found*, *Found (N copies; using first)*, or *No match*.

To relink:

1. Click **Search Folder…** and choose the folder where the media now lives. ATTACCA searches it and all its subfolders, matching files by name.
2. Check the green proposed paths and the summary line (e.g. *3 of 3 files matched.*).
3. Click **Apply Relink**. All matches are applied as **one** operation — a single `Ctrl+Z` reverses the whole relink. Nothing is played or fired by relinking.

Two things to know:

- There is **no menu entry** to open this dialog manually. If you click **Cancel**, get it back by re-opening the workspace (**File** → **Open…** and pick the same file).
- The dialog is **suppressed in Show Mode** — mid-show, missing media is only logged, never a pop-up.

Undo and redo

`Ctrl+Z` / `Ctrl+Y`, up to **200 steps**.

Undoable: every structural edit (create, delete, cut, paste, duplicate, grouping, marker changes, media relinks) and essentially every property you change in the inspector — levels, fades, geometry, light channels, triggers, notes, colors, all of it.

Edits are **coalesced** sensibly: dragging a fader or typing in a field produces one undo step, not fifty; releasing the slider or leaving the field starts the next step. Editing one property across a multi-selection is one step that restores every cue at once.

Never undoable (by design): playback and transport (GO, stop, pause, Panic), playhead and selection moves, the Edit/Show mode switch, and Preferences. Undo rewrites your *show*, never your *performance* — `Ctrl+Z` during a show can't yank back a running cue.

Finding cues

Press `Ctrl+F` to open the search bar. Search matches a cue's **number, name, notes, or media file path**, case-insensitively, as you type.

- `Enter` jumps to the next match, `Shift+Enter` to the previous (both wrap around).
- Matches inside collapsed groups work — the group expands automatically and the cue is selected.
- `Esc` closes the bar. While the search box has focus, `Space` types a space (it does not fire GO) — close the bar first if you want to fire.

> *[Illustration to be added in a future revision]*

Edit Mode and Show Mode

ATTACCA is always in one of two modes.

Edit Mode is for building: everything is editable, the inspector is live, and file operations are available.

Show Mode is for performing. It locks the show against accidents:

- The entire Edit menu (undo, cut/copy/paste, delete, Preferences) is disabled.
- All cue and cue-list editing is blocked; the inspector is disabled.
- File → New/Open/Save are unavailable.
- The Relink Media dialog stays suppressed.

What still works is everything you need to *run* the show: GO, stop, pause/resume, Panic, playhead movement, hotkeys, and triggers. Entering Show Mode also arms your wall-clock triggers — it's the "we're live" switch.

Switching modes: click the **Edit** / **Show** toggle buttons at the bottom-left of the status bar, use **View → Edit Mode**, or press `Ctrl+Shift+]` (enter Show Mode) / `Ctrl+Shift+[` (back to Edit Mode) / `Ctrl+E` (toggle).

You always know which mode you're in: in Show Mode the title bar reads `[Show]`, a red **SHOW MODE** badge lights up in the status bar, and a red tint washes over the toolbar row.

> *[Illustration to be added in a future revision]*

The Show Mode password

To keep a wandering hand (or a curious volunteer) from unlocking the show, set an exit password: **Preferences → General → Show Mode Security**, check **Require password to exit Show Mode**, and enter the password twice. It's stored securely (as a hash) inside the workspace — not readable from the file.

With a password set, leaving Show Mode pops a dialog titled **Enter Password to Return to Edit Mode**. A wrong entry shows *Incorrect password* and the show stays locked; **Cancel** stays in Show Mode.

> *[Illustration to be added in a future revision]*

If you've genuinely lost the password, see the recovery note in *Troubleshooting* — you will never be locked out of your own show.

Common issues

- **Everything is greyed out / I can't edit anything** — you're in Show Mode. Check for the red **SHOW MODE** badge and switch back with the **Edit** toggle (password if one is set).
- **The Relink dialog appeared, I cancelled it, and now I can't find it** — there's no menu entry; re-open the workspace to trigger it again. It also never appears in Show Mode.
- **Pressing `Space` types a space instead of firing GO** — the search bar (or another text box) has focus. Press `Esc` to close/leave it, then GO.
- **No recent files in the File menu** — by design; recents live on the Welcome window, shown whenever you start ATTACCA without a file.
- **The title bar doesn't show an asterisk, so I can't tell if I've saved** — ATTACCA doesn't mark dirty state in the title; it will always prompt *Save changes to the current workspace?* before closing unsaved work. When in doubt, `Ctrl+S`.
- **I moved my show folder and every media cue still works — is that right?** — yes: media paths are stored relative to the workspace, so moving the whole folder is the supported workflow.
- **"This workspace appears to be open in another ATTACCA instance" on open** — another copy of ATTACCA (possibly on another machine via a shared drive) has the file open. Don't run two instances against one file; close the other one first.
- **Video/camera cues won't fire right after several crashes** — the app is in Safe Mode (entered silently after 3 crashes in an hour). Save your work, restart ATTACCA, and if crashes persist grab the logs from `%APPDATA%\ATTACCA\logs`.

Working with Cues

Everything in an ATTACCA show is a cue: a sound, a video, a lighting look, a MIDI message, a note to yourself. This chapter covers the skills you will use on every cue of every type — creating, selecting, naming, numbering, reordering, editing in the inspector, sequencing with waits and auto-continues, grouping, and understanding the armed / flagged / broken / locked states.

Creating cues

ATTACCA has no "Cue" or "Add" menu. Cues are created from exactly two surfaces:

1. **The cue toolbar** — the row of buttons directly under the transport area, one button per cue type.
2. **Keyboard shortcuts** — the ten most common cue types have default `Ctrl+0` through `Ctrl+9` shortcuts. Every shortcut is remappable in **Preferences → Keyboard Shortcuts**.

There is no drag-and-drop creation either: dropping an audio or video file from Explorer onto the cue list does **not** create a cue. Create the cue first, then assign its file in the inspector.

> **Why it works this way:** Keeping creation on one always-visible toolbar means the full cue-type palette is one click away and never buried in a menu tree. During programming you will mostly live on `Ctrl+1` (Audio), `Ctrl+3` (Video), and `Ctrl+6` (Light).

[Illustration to be added in a future revision]

The cue toolbar, button by button

The toolbar is organized into six groups separated by thin vertical dividers. The buttons are text-only, in this exact order:

Group	Button	Creates	Default shortcut
Audio	**Group**	Group cue (container for other cues)	`Ctrl+0`
Audio	**Audio**	Audio file playback cue	`Ctrl+1`
Audio	**Mic**	Mic cue — *button visible but disabled; not available in this version (see the Mic section in* Media Cues*)*	—
Video	**Video**	Video file playback cue	`Ctrl+3`
Video	**Camera**	Live camera feed cue	`Ctrl+4`
Video	**Image**	Static image with hold + fade in/out	—
Video	**Text**	Styled text on a video display	`Ctrl+5`
Lighting	**Light**	DMX lighting look	`Ctrl+6`

Group	Button	Creates	Default shortcut
Lighting	**Fade**	Fades another cue's levels/geometry over time	`Ctrl+7`
Lighting	**Effect**	Chase / sine / color cycle running on top of cue state	—
Communication	**Network**	OSC / plain-text / hex network message — *button visible but disabled; not available in this version*	—
Communication	**MIDI**	MIDI voice / MSC / SysEx message	`Ctrl+9`
Communication	**MIDI File**	Plays a Standard MIDI File	—
Communication	**Timecode**	Transmits MTC or LTC timecode	—
Transport	**Start**	Starts a target cue	—
Transport	**Stop**	Stops a target cue	—
Transport	**Pause**	Pauses a target cue	—
Transport	**Load**	Loads a target cue	—
Transport	**Reset**	Resets a target cue	—
Transport	**Devamp**	Exits a target cue's loop — *button visible but disabled; not available in this version*	—
Transport	**GoTo**	Moves the playhead to a target cue	—
Transport	**Target**	Retargets another cue — *button visible but disabled; not available in this version*	—
Transport	**Arm**	Arms a target cue	—
Transport	**Disarm**	Disarms a target cue	—
Utility	**Wait**	Delays an auto-continue chain	—
Utility	**Memo**	Does nothing — a labeled note in the list	—
Utility	**Script**	Runs a Lua script	—
Utility	**Serial**	Sends an RS-232 serial command	—
Utility	**HTTP**	Sends an HTTP/REST request	—
Utility	**OSC**	Sends a single OSC message via UDP/TCP	—
Utility	**Timer**	Controls the Show Timer window	—

That is all 31 cue types — every type in ATTACCA has a toolbar button. Types without a default shortcut are toolbar-only (you can assign them shortcuts in **Preferences → Keyboard Shortcuts**). The **Mic**, **Network**, **Devamp**, and **Target** buttons are disabled in this release — those four cue types are not available in this version; `Ctrl+2` and `Ctrl+8` — which used to create Mic and Network cues — are intentionally unassigned and do not appear in the Keyboard Shortcuts list.

One heads-up: the "(Ctrl+N)" hints in the toolbar tooltips are fixed text. If you remap a creation shortcut, the toolbar tooltip will still show the original default.

Cue types by tier

Free covers Audio, Group, Fade, Wait, Memo, Timer, and the transport types (Start, Stop, Pause, Load, Reset, GoTo, Arm, Disarm).

Standard adds the video family: Video, Camera, Image, and Text.

Pro adds Light, Effect, MIDI, MIDI File, Timecode, Script, Serial, HTTP, and OSC.

Clicking the toolbar button for a type above your tier opens the upgrade dialog instead of creating the cue. The generated, feature-by-feature table is in *Licensing & Activation*.

(Mic, Devamp, Target, and Network are not available in this version at any tier — their buttons are disabled.)

Where new cues land

When you create a cue:

- If the selected cue is inside a Group, the new cue is inserted **inside that group**, right after the selection.
- Otherwise the new cue is inserted at the playhead position (or at the end of the list if the playhead is at the end).
- The cue number is assigned automatically, the playhead advances to the new cue, and the new cue becomes the selection.
- The default name is the type name plus "Cue" (for example "Audio Cue") until you rename it.
- Creation is undoable (`Ctrl+Z`).

Cue creation only works in Edit Mode — the toolbar does nothing in Show Mode.

Selecting cues

- **Click** a cue to select it. Clicking also moves the playhead to that cue — but never fires it. Only **GO**, `Spacebar` , or a trigger fires a cue.
- **Shift+Click** selects a range.
- **Ctrl+Click** adds or removes individual cues from the selection.
- `Ctrl+A` selects every cue in the list.

With multiple cues selected, the inspector shows only the **Basics** and **Triggers** tabs, its header reads "N cues selected", and any field you edit applies to every selected cue in a single undoable step. Fields whose values differ across the selection show an italic *(multiple values)* overlay until you type a new value.

To move the playhead to a cue without selecting-and-clicking (for example, from a long way away), right-click the cue and choose **Go To Cue**.

[Illustration to be added in a future revision]

Renaming a cue and editing its number

Names and numbers are edited in the inspector's **Basics** tab (there is no in-list editing):

1. Select the cue.
2. In the **Basics** tab, type into the **Name** field. Clearing the name reverts to the type's default name.
3. Type into the **Number** field to change the cue number. Numbers are free text — `2.5`, `A1`, and `Intermission` are all valid. Changing a number does not move the cue; it is a label, not a sort key.

Renumbering a whole list

The **Renumber Cues** dialog reassigns numbers to the cues in sequence:

1. Press `Ctrl+R`. This is the **only** way to open the dialog — it has no menu item. (The shortcut is listed as "Open Renumber Cues…" in **Preferences → Keyboard Shortcuts** if you want to remap it.)
2. Set **Start Number** (default `1`) and **Increment** (default `1`).
3. Check the live preview line, then click **Renumber** (or **Cancel**).

Renumbering is a single undoable action.

[Illustration to be added in a future revision]

Reordering cues

Reordering is drag-and-drop (Edit Mode only):

- Drag a cue and drop it on the **top half** of another row to insert it above, or the **bottom half** to insert it below.
- Drop on the **middle third of a Group row** to move the cue **into** that group (the group expands automatically).
- Drop **below the last row** to move the cue to the end of the list (this also pulls a cue out of a group).
- Multi-selection drags work: all selected cues move together, keeping their order.

There are no move-up/move-down keyboard shortcuts in this release — use drag, or cut and paste.

[Illustration to be added in a future revision]

Cut, copy, paste, duplicate, delete

Action	Shortcut	Also available
Cut	`Ctrl+X`	**Edit** menu
Copy	`Ctrl+C`	**Edit** menu, right-click menu
Paste	`Ctrl+V`	**Edit** menu, right-click menu
Duplicate	`Ctrl+D`	**Edit** menu, right-click menu

Action	Shortcut	Also available
Delete	`Del`	**Edit** menu, right-click menu
Undo / Redo	`Ctrl+Z` / `Ctrl+Y`	**Edit** menu

Notes:

- Multi-cue copy/paste works; a five-cue paste undoes in one `Ctrl+Z`.
- Deleting has no confirmation prompt — it is instant, but always undoable.
- The undo stack holds 200 steps and covers structural operations *and* essentially every inspector property edit.
- Pasting a cue type above your license tier is blocked the same way as creating one.
- All of these are Edit-Mode-only (the entire **Edit** menu is disabled in Show Mode).

Copy Effects (`Ctrl+Shift+C`) and **Paste Effects** (`Ctrl+Shift+V`) copy an audio cue's effect chain between cues — see *Audio in Depth*.

The cue-list right-click menu

Right-clicking a cue opens this menu (items marked † are disabled in Show Mode):

1. **Go To Cue** — moves the playhead to this cue without firing it
2. **Copy** `Ctrl+C`
3. **Paste** `Ctrl+V` †
4. **Duplicate** `Ctrl+D` †
5. **Copy Effects** `Ctrl+Shift+C`
6. **Paste Effects** `Ctrl+Shift+V` †
7. **Delete** `Del` †
8. **Arm/Disarm** † — toggles the armed state of the selection
9. **Group Selected Cues** `Ctrl+G` — wraps the current selection in a new Group cue
10. **Color** † — submenu of 21 swatched items, in this order: **None**, **Red**, **Orange**, **Yellow**, **Green**, **Blue**, **Purple**, **Crimson**, **Hot Pink**, **Berry**, **Indigo**, **Sky Blue**, **Cyan**, **Forest**, **Olive**, **Midnight**, **Peach**, **Lavender**, **Plum**, **Gray**, **Magenta**

The menu contains no "New Cue" items — creation lives on the toolbar.

> *[Illustration to be added in a future revision]*

The inspector: Basics tab

Every cue type — all 31 — has a **Basics** tab and a **Triggers** tab in the inspector. (Toggle the inspector with `Ctrl+I`; it is available in Edit Mode only.)

The **Basics** tab contains:

Field	What it does
Number	The cue number (free text).
Duration	How long the cue's action runs. Accepts `H:MM:SS.ff`, `M:SS.ff`, `S.ff`, or plain seconds; empty means 0. For media cues the duration is set automatically from the file. For a Light cue with a recording attached it is the take's playback length (shown with ∞ in the cue list when the take loops) and is edited by trimming the take, not here; for a fixture-based Light cue it is the fade time.
Pre Wait	Delay between firing the cue and its action starting. Same time formats.
Post Wait	Delay used by auto-continue — see "Waits, sequencing, and GO" below.
Continue	Combo box with three options, shown exactly as **DoNotContinue**, **AutoContinue**, **AutoFollow**.
Name	The cue's display name.
Target	Only shown for cue types that target something. File-target cues (Audio, Video, MIDI File) show the file path read-only here — the actual **Browse…** button is on the type's own I/O or Settings tab. Cue-target cues (Start, Stop, Pause, Load, Reset, Devamp, GoTo, Target, Arm, Disarm, Fade) show a combo box listing every cue in the workspace by number and name.
Color	Row color — the same 21 colors as the right-click Color submenu.
2nd	Enables a second color, drawn as an alternating stripe with the first (tooltip: "Enable second color for alternating stripe").
Armed	Whether the cue will fire. See "Armed and disarmed cues" below.
Flagged	A bookmark flag on the cue. See "Flagged cues" below.
Notes	Free-text notes. The playhead cue's notes are also shown in the header notes area next to GO.

Two handy single-key shortcuts while a cue is selected: `C` cycles the **Continue** mode, and `F` toggles **Flagged**.

[Illustration to be added in a future revision]

The inspector: Triggers tab

The **Triggers** tab lets a cue fire itself from things other than GO. It has two columns: **Trigger Sources** on the left and **Behaviors** on the right. Full setup detail (device enabling, matching rules, arming) is in *Control & Integration* — here is the map:

Trigger Sources

- **Hotkey** — check the box, click the capture field ("Click and press a key to assign hotkey"), and press a key. That key now fires this cue directly, from anywhere, without moving the playhead. `Esc`, the `Up` / `Down` arrows, and bare `Ctrl` / `Alt` / `Shift` are reserved and cannot be assigned. Multiple cues may share one hotkey.
- **MIDI Trigger** — check the box, then either click **Learn** (it changes to **Listening…** while capturing, and requires a MIDI input device to be enabled in Preferences) and send the message from your controller, or

fill the fields manually: **Type** (**Note On**, **Note Off**, **Control Change**, **Program Change**), **Ch** (0 = workspace default, 1–16 = specific), **Note** (name or number, with a **Note**/**Number** display toggle), and **Vel** (`any` , an exact value like `64` , or a comparison like `>0`). **Copy** and **Paste** buttons move a trigger configuration between cues.

- **Wall Clock** — fires the cue at a time of day, entered as 12-hour **Hour** / **Min** / **Sec** plus **AM**/**PM**. A note under the fields shows which timezone the trigger uses. Fires once per day, within a 5-second forward window.
- **Timecode** — *not available in this version*: the position field (`HH:MM:SS:FF`) is accepted and saved, but ATTACCA 1.0 does not chase incoming timecode, so this trigger never fires (see *Control & Integration*).
- **Audio Marker** — fires the cue when another audio cue's playback crosses a marker: pick the **Source** (the audio cue) and the **Marker**. A preview line summarizes the trigger.

Behaviors

- **Fade & Stop Others** — when this cue fires, fade out and stop other running cues. Scope options (shown exactly as in the app): **Disabled**, **Peers**, **List**, **All**, plus a fade time in seconds.
- **Duck/Boost audio of other cues** — checkbox plus **Level** (dB — negative ducks, positive boosts) and **Time** fields. See *Audio in Depth* for exactly how ducking behaves in this release before relying on it.
- **Second Trigger** — what happens if this cue is fired again while already running: **DoNothing**, **Panic**, **Stop**, **HardStop**, or **HardStopAndRestart**, plus a **Second trigger on release** checkbox (act on key-release instead of key-press).

Editing any trigger automatically arms the trigger system so you can test it immediately (see *Control & Integration* for the full arming model).

> *[Illustration to be added in a future revision]*

Waits, sequencing, and GO

What GO does

Pressing **GO** (or `Spacebar`):

1. Fires the cue at the playhead. Disarmed cues are skipped — GO fires the next *armed* cue at or after the playhead.
2. Advances the playhead to the next armed cue.

At the end of the list, GO does nothing — it never wraps back to the top. Click a cue (or use the arrow keys) to reposition the playhead.

There is **no GO lockout in this release**: pressing GO twice in quick succession fires two consecutive cues. Rely on discipline, not the software, when hovering over the spacebar.

Cue timeline: pre-wait, action, post-wait

Every cue runs through the same phases: **pre-wait** → **action** → **complete**. The cue list shows live countdowns in the **Pre-Wait**, **Duration**, and **Post-Wait** columns while each phase runs.

- **Pre Wait** delays the action. Fire a cue with a 2-second pre-wait and nothing happens for 2 seconds, then the action starts.
- **Post Wait** matters only for auto-continues, and — this is the important part — it counts **from the moment the action starts**, not from when it ends.

The three continue modes

The **Continue** combo in Basics controls what happens after a cue, using these exact labels:

Label	Meaning
DoNotContinue	Nothing happens automatically. The operator presses GO for the next cue. This is the default.
AutoContinue	The **next** cue fires automatically when **this cue's post-wait elapses**, counted from this cue's action start. Post Wait `0` = the next cue starts at the same instant as this one. Post Wait `3` = the next cue starts 3 seconds after this one starts.
AutoFollow	Set on the cue that should **wait**: a cue set to AutoFollow fires automatically when the **previous** cue's action completes.

Note the difference in where you set them: **AutoContinue goes on the earlier cue** ("when I start, start the next one after my post-wait"); **AutoFollow goes on the later cue** ("start me when the cue before me finishes").

Auto-continue and auto-follow chains skip disarmed and broken cues, and they do **not** move the playhead — the playhead only advances on GO. This matches standard show-control practice: a GO can kick off a long automatic sequence while the playhead stays parked on the next thing the operator must fire by hand.

The rightmost cue-list column (header ▶) shows a yellow glyph indicating each cue's continue mode at a glance. With a cue selected, tap `C` to cycle through the three modes without touching the inspector.

[Illustration to be added in a future revision]

Targeting: what cues point at

Different cue types target different things:

Target	Cue types	Where you set it
A media file	Audio, Video, MIDI File	**Browse…** on the cue's **I/O** (or **Settings**) tab; the path also shows read-only in Basics
Another cue	Start, Stop, Pause, Load, Reset, Devamp, GoTo, Target, Arm, Disarm, Fade	The **Target** combo in Basics — pick any cue in the workspace
An image file	Image	The **…** browse button on the Image cue's **I/O** tab (not the Basics row)
A camera device	Camera	The **Device:** combo on the Camera tab — a device, not a file
A display	Video, Camera, Image, Text	Each type's **Display:** combo on its own tab

Target	Cue types	Where you set it
Fixtures / DMX channels	Light, Effect	The Levels / Effect tabs (see *Lighting*)
Nothing	Group, Mic, Wait, Memo, Script, Serial, HTTP, OSC, Timer, Network, MIDI, Timecode	These carry their content in their own inspector tabs

Groups

A Group cue is a container for other cues.

Creating a group

- Click **Group** on the toolbar (or `Ctrl+0`) to create an empty group, then drag cues into it (drop on the middle of the group's row), **or**
- Select the cues you want to contain, right-click, and choose **Group Selected Cues** (`Ctrl+G`) — the selection is wrapped in a new group in place.

Group modes

The Group cue's inspector adds a **Group** tab with a **Mode** combo and a read-only **Children** count. The mode labels appear in the app exactly as written here (they are internal names, shown as-is):

Mode	What firing the group does
List	Plays the children **in order, one after another** — each child starts when the previous one completes. The group runs until the last child finishes.
StartFirstEnter	Starts the **first child only**, and moves the playhead to the second child inside the group — so each GO steps through the group one child at a time.
StartFirst	Starts the **first child only**; the playhead skips past the group to the next top-level cue.
Timeline	Starts **all children at once**, with each child's **Pre Wait** acting as its offset from the group's start. Build timed sequences by giving children staggered pre-waits.
StartRandom	Fires **one randomly chosen child**. Round-robin: a child will not repeat until every other child has had a turn.
Cart	Turns the children into a clickable button grid (a cue cart) in the main window. Firing the group itself from the list does nothing — carts are played by clicking cells. See the cart view in *The Interface*.
Playlist	Plays children sequentially like List, with two extra **Playlist Options** checkboxes that appear in Playlist mode only: **Shuffle** and **Loop**.
StartAll	Starts **all children simultaneously**. Unlike Timeline, a child's Pre Wait is just an ordinary delay, not a designed timeline offset.

Disarmed and broken children are skipped in every mode.

Collapsing and expanding

Group rows show a ▶/▼ toggle at the left of the name — click it to collapse or expand the group. Children display indented under the group. Press `.` (period) to expand all groups and `,` (comma) to collapse all. Searching with `Ctrl+F` reaches into collapsed groups and expands them when a match is found inside.

[Illustration to be added in a future revision]

Cue states: armed, flagged, broken, locked

Armed and disarmed cues

Every cue has an **Armed** checkbox in Basics (default on). Disarmed cues:

- render at reduced opacity (dimmed) in the list;
- are **skipped** by GO, by auto-continue/auto-follow chains, and by group playback;
- refuse to fire from any trigger (hotkey, MIDI, wall clock, marker).

Toggle arming from the right-click menu (**Arm/Disarm**) or with `Ctrl+Shift+A`. Disarming is the standard way to take a cue out of the running order without deleting it.

Flagged cues

Flagged (Basics checkbox, or tap `F` with the cue selected) is a per-cue bookmark you can use however you like — commonly "needs attention before opening night." In this release the flag does not draw a marker in the cue list; it is visible in the Basics tab and stored with the workspace.

Broken cues

A cue is *broken* when it cannot perform its action — most commonly a media cue whose file is missing, or a Mic cue loaded from an older workspace (Mic cues can no longer be created — the toolbar button is disabled — and any that load from a legacy workspace are always broken in this release; see *Media Cues*).

Be aware: **the cue list does not draw a special broken icon.** A broken cue looks like any other cue in the list. What you will actually observe:

- If media files are missing when a workspace loads, ATTACCA offers the **Relink Media** dialog automatically (see *The Interface*) — that is your primary missing-media surface.
- Pressing GO on a broken cue "fires" it as a silent no-op and advances the playhead by exactly one, so your GO count stays predictable mid-show.
- Auto-continue chains and triggers skip broken cues entirely.

If a cue seems to do nothing when fired, check its file target in the inspector first.

Locked cues

If you open a workspace containing cue types above your license tier, those cues appear **locked**: the row renders at half opacity with a grey padlock glyph in the status column, and the padlock's tooltip names the tier that unlocks it — "This cue requires ATTACCA Standard" or "This cue requires ATTACCA Pro". Locked cues

cannot be fired or edited until that tier is licensed. Nothing is deleted — the cues sit intact and unlock the moment the license activates. See *Licensing & Activation*.

[Illustration to be added in a future revision]

Worked example: a three-cue sequence with auto-continues

Goal: one GO brings up the lights, starts a music sting 2 seconds later, and shows a title card the moment the music ends.

Build this order in the list:

#	Cue	Continue	Post Wait
1	`Lights up` (Light)	**AutoContinue**	`2`
2	`Sting` (Audio)	**DoNotContinue**	—
3	`Title card` (Image)	**AutoFollow**	—

1. Click **Light** on the toolbar (or `Ctrl+6`). In **Basics**, name it `Lights up`, then build its look on the **Levels** tab (see *Lighting*).
2. With `Lights up` selected, set **Continue** to **AutoContinue** and **Post Wait** to `2`. This means: "2 seconds after I start, fire the next cue."
3. Click **Audio** (`Ctrl+1`). Name it `Sting`; on the **I/O** tab click **Browse…** and pick your music file. Leave its **Continue** at **DoNotContinue** — the cue after it will do the watching.
4. Click **Image**. Name it `Title card`; on its **I/O** tab browse to your title image and set its **Display:**. Set its **Continue** to **AutoFollow** — this means: "start me when the cue directly above me finishes." Since the cue above is `Sting`, the card appears exactly when the music ends.
5. Move the playhead to `Lights up` and press `Spacebar`. The lights come up, the Post-Wait column counts down 2 seconds, the sting starts, and — however long the sting runs — the title card fires itself at the end. The playhead advanced only one step and is waiting on `Sting`'s row; the chain runs on its own.
6. Press `Esc` at any time to panic-stop the whole chain.

Notice how the two automatic modes divide the work: **AutoContinue on the earlier cue** handles "start the next thing N seconds after me"; **AutoFollow on the later cue** handles "start me when the thing before me ends."

[Illustration to be added in a future revision]

Common issues

- **"I can't find the New Cue menu."** There isn't one. Cues are created only from the toolbar under the transport row, or with `Ctrl+0` – `Ctrl+9`.
- **Dragging a file from Explorer onto the cue list does nothing.** By design — create the cue first, then assign the file with **Browse…** on its I/O tab.

- **GO skipped my cue.** The cue is disarmed (dimmed row). Re-arm it via right-click → **Arm/Disarm** or the **Armed** checkbox in Basics.
- **GO did "nothing" on a cue, but the playhead moved.** The cue is broken — usually a missing file. Check the file target on its I/O tab; if the whole show moved folders, reopen the workspace and use the **Relink Media** dialog.
- **My auto-continue fires at the wrong time.** Remember Post Wait counts from when the cue **starts**, not when it ends. If you want "start when the previous cue ends," set **AutoFollow** on the *following* cue instead.
- **The next cue fired itself but the playhead didn't move.** Correct behavior — auto-continue chains never move the playhead. The playhead marks the next *manual* GO.
- **I renumbered but the cues didn't reorder.** Numbers are labels, not sort keys. Reorder by dragging; use `Ctrl+R` (Renumber Cues) to make numbers match the new order.
- **The toolbar button shows `Ctrl+1` but my remapped key is different.** Toolbar tooltips show the factory defaults and don't update after remapping. Your remapped shortcut still works.
- **Nothing in the toolbar responds.** You are in Show Mode — all editing (creation, deletion, reordering, inspector) is locked. Toggle back with the **Edit** button in the status bar.
- **Clicking a cue fired it!** It didn't — clicking only moves the playhead and selection. If something started, check for a hotkey trigger assigned to a key you pressed, or a wall-clock/timecode trigger that came due.

Media Cues: Audio, Mic, Video, Camera, Image, Text

This chapter covers the six cue types that put sound and pictures in front of an audience. For each type you'll find how to create it, what it targets, every inspector tab and field, and a short worked example. Deeper subjects — loudness and effects theory, display mapping, output windows — live in *Audio in Depth* and *Video & Camera in Depth*.

A quick map before diving in:

Cue type	Toolbar button	Shortcut	Tier	Plays
Audio	**Audio**	`Ctrl+1`	Free	An audio file
Mic	**Mic**	—	—	*Not available in this version — toolbar button disabled*
Video	**Video**	`Ctrl+3`	Standard	A video file on a display
Camera	**Camera**	`Ctrl+4`	Standard	A live camera feed on a display
Image	**Image**	—	Standard	A still image on a display
Text	**Text**	`Ctrl+5`	Standard	Styled text on a display

The video family — Video, Camera, Image, and Text — is included in **ATTACCA Standard** and **Pro**. See *Licensing & Activation* for the full tier table.

Audio cues

An Audio cue plays a sound file — music, effects, playback stems — through your chosen output device, with trimming, looping, rate control, fades, markers, and an optional effects chain.

Create: click **Audio** on the toolbar or press `Ctrl+1`. Available on the Free tier.

Target: an audio file. The path shows read-only in **Basics**, but you assign it on the **I/O** tab with **Browse…**. The file dialog (titled **Select File Target**) filters for: audio files `.wav .mp3 .aiff .aif .flac .ogg .wma .m4a`, plus video files `.mp4 .mov .avi .wmv .mkv` (an Audio cue can play a video file's soundtrack), MIDI files `.mid .midi`, and All files. WAV is the safest choice for zero-decode-latency playback.

Inspector tabs: Basics, **Triggers**, **I/O**, **Time & Loops**, **Levels**, **Effects**.

I/O tab

Field	What it does
File:	The current audio file (shows `(no file)` when unset).
Browse…	Opens the file dialog and sets the target. The cue's Duration is probed from the file automatically.

Field	What it does
Format:	Read-only format line derived from the file (e.g. `WAV file`).
Patch:	Audio output patch number; `0` = unpatched (plays on the default device).
Output:	Read-only — `(no patch assigned)` or `Patch N`.
Device:	Read-only — the audio device this cue will use.

[Illustration to be added in a future revision]

Time & Loops tab

Field	Range / format	What it does
Start time: / **End time:**	`MM:SS.mmm` or seconds	Trim points — playback runs from Start to End. Editing Start moves the `S` marker on the waveform.
Play __ time(s)	count ≥ 1	Plays the trimmed region exactly that many times.
Infinite loop	—	Loops until stopped.
Rate:	0.03 – 33.0	Playback speed multiplier. Changes apply live to a running cue.
Preserve pitch	checkbox	Keeps pitch constant while the rate changes (slightly more CPU). Off = classic varispeed, where speed changes shift pitch.
Integrated fade	checkbox	Enables the cue's built-in fade envelope, revealing **In:** and **Out:** times (seconds) and **Lock fade to cue**.
Slices:	read-only	Shown only when the cue has slice markers.
Zoom	1 – 20x	Waveform zoom (slider, or `Ctrl+Wheel` on the waveform).

The waveform. Below the fields is the interactive waveform display:

- **Click** anywhere to seek the playhead — if the cue is playing, it jumps live. Seeking does **not** fire audio-marker triggers; only natural playback crossing does, so you can safely click around during rehearsal.
- **Ctrl+Click** adds a marker at that point.
- **Shift+Click** adds a slice marker.
- **Drag** a marker to move it (one undo step per drag).
- **Right-click** a marker for **Rename Marker** / **Delete Marker** / **Go to Position**; right-click a slice for **Delete Slice Marker**; right-click empty waveform for **Add Marker Here**.
- **Ctrl+Wheel** zooms 1–20x.

Multi-channel files draw every channel as a stacked lane.

The Markers section. Under the waveform, the **+ Marker** button adds a marker at the current playhead position, and a grid lists all markers with editable **Name** and **Time** (`MM:SS.xx`) columns and a delete **X** per row. Markers do two jobs: navigation landmarks, and **trigger sources** — any other cue can fire when playback

crosses one (its Triggers tab → **Audio Marker**). The cue list's **Marker** column shows a live countdown to the next marker that will fire something.

[Illustration to be added in a future revision]

Levels tab

Control	What it does
Main Level	Vertical fader plus dB text box, −60 to +12 dB. The text box also accepts `-INF` for silence. **0.0 dB is unity** — the file plays back exactly at its recorded level. Changes apply live while the cue plays. If unity sounds different than other software on your machine, see the loudness explanation in *Audio in Depth*.
Set Default Levels	Resets main and all outputs to 0 dB.
Set All Silent	Sets main and all outputs to −INF.
Channels: / Outputs:	Read-only info — the file's true channel count (probed from the file) and the output count.
Clear Mutes / Clear Solos	Present but permanently disabled in this release — placeholders for the matrix mixer.

The right side of the tab reads "Per-output levels are coming in a future update." — and that is the honest state of things: **there is no per-output matrix mixer in this release.** Use **Main Level** to control the cue's volume, and the workspace-level output device selection for routing.

[Illustration to be added in a future revision]

Effects tab

An opt-in per-cue effects chain — a fresh Audio cue has no effects until you add them. Eight effects are available via the add buttons **+ EQ**, **+ Reverb**, **+ Comp**, **+ Delay**, **+ Chorus**, **+ Echo**, **+ Gate**, **+ Limiter** (full names: Parametric EQ, Reverb, Compressor, Delay, Chorus, Echo, Gate, Limiter). Each effect panel has an enable checkbox, save/load preset buttons, reorder arrows, and a remove button; **Save Chain** / **Load Chain** store whole chains as presets. Effects can be edited live during playback.

Audio effects are part of the Free tier — no license needed.

Parameter-by-parameter detail, plus the automatic protective limiting ATTACCA adds behind the scenes, is in *Audio in Depth*.

[Illustration to be added in a future revision]

Worked example: pre-show music

1. Press `Ctrl+1`, name the cue `Preshow`.
2. **I/O** tab → **Browse…** → pick your playlist track.
3. **Time & Loops** → select **Infinite loop**.

4. **Levels** → set **Main Level** to `-12` so announcements can sit on top.
5. Fire it with GO; when the house opens, it loops forever. Stop it before the show with a Stop cue targeting `Preshow` (or select it and press `S`).

Audio gotchas

- **Duration is probed from the file** on assignment. If you trim with Start/End, the countdowns follow the trimmed length.
- Rate and Main Level changes are **live** — great for rehearsal, so remember to undo experiments (`Ctrl+Z` covers inspector edits).
- Multi-channel files play, and the waveform shows every channel — but there is **no channel routing** in this release; all channels go to the cue's device as-is.
- The **Duck/Boost** behavior on the Triggers tab has sharp edges in this release — read *Audio in Depth* before using it in a show.

Mic cues

Be direct about this one: **the Mic cue is not available in this version.** The **Mic** toolbar button is visible but disabled, and the `Ctrl+2` shortcut it once had is unassigned. Mic cues you'll encounter come only from workspaces saved by earlier builds — such a cue will not play, and GO passes over it as a no-op. It loads normally, its inspector shows only **Basics** and **Triggers**, and nothing bad happens; there is simply nothing to configure.

For live sound reinforcement, route your microphones through your mixing console — ATTACCA's job is playback, not live input.

Video cues (Standard)

A Video cue plays a movie file on a chosen display, with geometry (position/scale/rotation/crop), color adjustment, looping, rate control, and built-in fades. Video output is software-decoded and always plays fullscreen on its target display's output window; geometry positions the picture within that stage.

Create: click **Video** on the toolbar or press `Ctrl+3`. Requires Standard or Pro.

Target: a video file, assigned on the **I/O** tab. The file dialog (titled **Select Video File**) filters for video files `.mp4` `.mov` `.avi` `.mkv` `.wmv` `.webm` `.m4v`, image files `.png` `.jpg` `.jpeg` `.bmp` `.tiff` `.tif` `.gif`, and All Files. H.264 in an MP4 container is the recommended safe format; see *Video & Camera in Depth* for the full codec list.

Inspector tabs (7): Basics, **Triggers**, **I/O**, **Audio**, **Geometry**, **Time & Loops**, **Effects**.

I/O tab

Field	What it does
File: / **Browse…**	The video file target.

Field	What it does
Format:	Read-only summary: extension, resolution, frame rate, length.
Preview	A first-frame thumbnail (appears once a file is set) that reflects scale, Z-rotation, opacity, and anchor before you ever GO, plus resolution/duration/codec/size lines.
Display:	Which display the video outputs on — a combo listing `Display N — Name` for every mapped display.
Display #:	The stage number for this output.
Fade In:	Seconds; `0` = instant. Fades opacity from 0 to the target on GO.
Fade Out:	Seconds; `0` = instant removal. Fades opacity to 0 on stop/replace — and leads into the end of the clip; see Time & Loops below.

If the assigned display is missing (unplugged projector, changed setup): the cue's display assignment is **preserved** — ATTACCA never rewrites your saved choice. At workspace load, output falls back to the primary/first display **for that session only**, and a warning is logged. Reconnect the display and reload the workspace to get your mapping back. Note that unplugging a display *mid-session* is not detected — see *Video & Camera in Depth* for the full fallback rules and known limitations.

[Illustration to be added in a future revision]

Audio tab

The video's own soundtrack:

- **Volume:** slider, 0–100%, applies live.
- **Mute** checkbox, applies live.

Geometry tab

Every numeric row is a slider plus a numeric box plus a ↺ reset button (typed values may exceed the slider's range).

Field	Range	Notes
Mode:	**FillStage** / **Custom**	FillStage sizes the picture to the display automatically; Custom reveals the Transform section.
Fill Style:	**Fit** / **Fill** / **Stretch**	Fit letterboxes, Fill crops to cover, Stretch distorts to fill.
Opacity:	0 – 1 (shown as %)	
Layer:	0 – 1000	Z-order when multiple visual cues share a display — higher is on top.
Blend Mode:	23 modes	Normal, Additive, Multiply, Screen, Overlay, Darken, Lighten, ColorDodge, ColorBurn, HardLight, SoftLight, Difference, Exclusion, Hue, Saturation, Color, Luminosity, LinearDodge, LinearBurn, VividLight, LinearLight, PinLight, HardMix.

Field	Range	Notes
Position X: / **Position Y:**	−5000 – 5000 px	Custom mode only.
Scale X: / **Scale Y:**	0.01 – 5.0	With a 🔒 button to lock X/Y together. Custom mode only.
Rotation: / **Rotation X:** / **Rotation Y:**	−360 – 360°	Z, X, and Y rotation. Custom mode only.
Anchor X: / **Anchor Y:**	0 – 1	Transform anchor, with preset buttons **Center**, **TL**, **TR**, **BL**, **BR**. Custom mode only.
Crop Top:/Bottom: and **Left:/Right:**	0–2160 / 0–3840 px	Custom mode only.
Preserve Aspect Ratio	checkbox	
Smooth (Bilinear Filtering)	checkbox	

Geometry is stored on the cue and re-applied on every play — retriggering a cue always starts from its saved geometry.

[Illustration to be added in a future revision]

Time & Loops tab

Field	Range	What it does
Start Time: / **End Time:**	`mm:ss.fff` or seconds	Trim in/out points.
Rate:	0.03 – 33.0	Playback speed.
Play Count:	≥ 1	Plays the clip exactly that many times (disabled while Infinite Loop is checked).
Infinite Loop	checkbox	Loops until stopped.
Hold Last Frame	checkbox	When the clip ends, freeze on its final frame instead of closing the output. The picture stays until the cue is stopped, panicked, or replaced.

How fade-out interacts with the end of the clip: the **Fade Out** time (I/O tab) *leads into* the end. ATTACCA starts the fade early so that it **completes exactly at the End Time** — the picture is fully dark at the moment the clip ends, rather than freezing on the last frame and then fading. Operationally: a 10-second clip with a 2-second fade-out starts fading at 8 seconds. The lead-in applies only on the final pass of a Play Count (mid-loop passes don't fade), and it does not arm at all when **Hold Last Frame** is checked — a held frame stays at full opacity until you stop the cue.

[Illustration to be added in a future revision]

Effects tab

Live color adjustment sliders:

Field	Range	Normal
Blur:	0 – 50	0 = off
Brightness:	−1 – +1	0
Contrast:	0 – 3	1
Saturation:	0 – 3	1 (0 = grayscale)

All four apply live to a playing cue. Set **Saturation** to 0 for an instant black-and-white treatment. There is **no gamma control** in this release.

[Illustration to be added in a future revision]

Worked example: a looping background with a clean button

1. Press `Ctrl+3`, name the cue `BG loop`.
2. **I/O** → **Browse…** → pick the loop file; set **Display:** to your projector; set **Fade In:** `1` and **Fade Out:** `2`.
3. **Time & Loops** → check **Infinite Loop**.
4. GO — the loop fades in over 1 second and runs. When the scene ends, stop it (Stop cue or `S`) and it fades out over 2 seconds.
5. Variation: for a one-shot video that should end on its final image (a logo, a tableau), uncheck Infinite Loop and check **Hold Last Frame** — the last frame stays up until your next visual cue replaces it.

Video gotchas

- Decoding is **software-only** — 4K/high-bitrate files cost real CPU. Test on the show machine, and prefer H.264 MP4 renders.
- **Play Count + Fade Out**: the fade happens once, on the final pass.
- A missing display never breaks the cue — it falls back to the primary display for the session. If video shows up on the wrong screen, check that the intended display is connected and remapped in **Preferences** → **Displays**, then reload.
- Panic (`Esc`) fades video over the panic fade time and closes the output window; `Ctrl+Shift+F12` is the global emergency “close all video output windows” hotkey.

Camera cues (Standard)

A Camera cue puts a live camera feed (webcam, USB capture card) on a display — IMAG, lobby cameras, conductor cams.

Create: click **Camera** on the toolbar or press `Ctrl+4`. Requires Standard or Pro.

Target: a capture device, not a file — picked on the **Camera** tab.

Inspector tabs (4): Basics, **Triggers**, **Camera**, **Geometry**.

Camera tab

Field	What it does
Device:	Editable combo of detected cameras (shows `Scanning...` while enumerating and `No cameras detected` if none). You can also type a device name for a camera that will be connected later.
(refresh button)	Re-scans for cameras.
Status:	A colored dot plus text — see "Camera warmup" below.
Resolution:	Editable combo — `1920x1080`, `1280x720`, `3840x2160`, `640x480`, or type any `WxH`.
FPS:	Editable combo — 15, 24, 25, 30, 50, 60.
Display: / **Display #:**	Output display and stage number, same as Video.
Fade In: / **Fade Out:**	Seconds; built-in opacity fades, same semantics as Video.

Camera warmup — why GO is instant

Opening a camera takes 2–3 seconds at the hardware level. ATTACCA hides that cost: **every camera referenced by any Camera cue is opened and kept warm from the moment the workspace loads.** GO simply switches the already-running feed to the visible output — the first frame lands in well under a tenth of a second.

The **Status:** indicator tells you where the device is in that lifecycle:

Status	Meaning
Idle	No warmup entry for this device yet.
Warming… (yellow)	Device opening in the background.
Ready (green)	Warm and streaming — GO will be instant.
Live	Adopted by a running Camera cue and visible.
Not connected (red)	Open failed — unplugged, in use by another app, or a driver error.

There is no on/off switch for warmup — it is automatic. Two practical consequences:

- **The camera's LED stays on from workspace load onward**, and the device consumes USB bandwidth as if live. That's the warm pool doing its job, not a stuck cue.
- The warm pool **survives panic** — after `Esc`, you can re-fire a camera cue instantly.

> **Why it works this way:** A visible 2–3 second black hole on GO is unacceptable in performance. Paying the open cost once at load, invisibly, means every camera GO in the show is instant.

One device, one output

Cues that use the **same physical camera share one warm feed** — the device is opened once. That also means one camera **cannot appear on two displays at the same time**: firing a second Camera cue that uses the same device moves the feed to the new output rather than duplicating it. Use two devices for two simultaneous outputs.

Geometry tab

Identical to the Video cue's Geometry tab, field for field — Mode (**FillStage**/**Custom**), **Fill Style**, **Opacity**, **Layer**, **Blend Mode** (23 modes), Position/Scale/Rotation/Anchor/Crop in Custom mode, **Preserve Aspect Ratio**, **Smooth (Bilinear Filtering)**.

[Illustration to be added in a future revision]

Worked example: IMAG on the side screens

1. Press `Ctrl+4`, name the cue `IMAG on`.
2. **Camera** tab → **Device:** pick your capture card; **Resolution:** `1920x1080`; **FPS:** 30; **Display:** the side-screen output; **Fade In:** `0.5`.
3. Wait for **Status:** to show **Ready** (it warms automatically after the device is assigned).
4. GO — the feed appears instantly with a half-second fade. Stop the cue to drop it.

Camera gotchas

- **Status: Not connected** usually means another application (Zoom, OBS, the manufacturer's utility) has the camera open. Close it and click the refresh button.
- The camera LED being lit before you ever fire the cue is **normal** — see warmup above.
- Same device on two displays: not possible — the newest cue takes the feed.
- No file, no duration: a Camera cue runs until stopped, panicked, or its display is taken over by another visual cue.

Image cues (Standard)

An Image cue puts a still image on a display with an optional timed hold and fade in/out — title cards, sponsor slides, blackout cards, projected scenery.

Create: click **Image** on the toolbar (tooltip: "Image Cue — static image with hold + fade in/out"). No default shortcut. Requires Standard or Pro.

Target: an image file, assigned on the **I/O** tab (not in Basics).

Inspector tabs (5): Basics, **Triggers**, **I/O**, **Geometry**, **Effects**.

I/O tab

Field	What it does
File: / **…**	The image file. The browse dialog (titled **Select Image File**) filters `.png .jpg .jpeg .bmp .webp .gif .tif .tiff .ico`. A hint line lists the formats: JPEG, PNG, BMP, WEBP, GIF, TIFF, ICO.
Display:	Output display combo.
Opacity:	0.0 – 1.0 — the target opacity the fade-in rises to.
Hold (s):	How long the image stays up. `0` **= indefinite** — it stays until stopped, panicked, or replaced.
Fade In: / **Fade Out:**	Seconds, `0` = instant.

Layer is set on the **Geometry** tab (deliberately not duplicated here).

Geometry and Effects tabs

Both are identical to the Video cue's tabs, field for field — the full transform surface (Mode, Fill Style, Opacity, Layer, Blend Mode, Position/Scale/Rotation/Anchor/Crop) and the four color sliders (Blur, Brightness, Contrast, Saturation). Everything works the same on a still image as on video.

[Illustration to be added in a future revision]

Worked example: a timed sponsor slide

1. Click **Image**, name it `Sponsor 1`.
2. **I/O** → **…** → pick the slide; **Display:** the lobby screen; **Hold (s):** `20`; **Fade In:** `1`; **Fade Out:** `1`.
3. Set **Continue** to **AutoContinue** in Basics with a suitable Post Wait and stack several of these to build a rotating slideshow, or set Hold to `0` and advance manually with GO plus a Stop cue.

Image gotchas

- **Hold 0 is not "no image"** — it means *forever*. For a timed slide, set the seconds you want.
- Animated GIFs load as stills — the first frame is displayed.

Text cues (Standard)

A Text cue renders styled text on a display — supertitles, warnings, speaker names, scene labels.

Create: click **Text** on the toolbar or press `Ctrl+5`. Requires Standard or Pro.

Target: none — the text itself lives in the cue.

Inspector tabs (3): Basics, **Triggers**, **Text**.

Text tab

Field	What it does
Text Content	Multiline box — the text to display.
Font:	Font family **typed by name** (default `Arial`). There is no font picker in this release — type the family name exactly as Windows knows it.
Size:	Point size, default 48.
Color:	Text color as a **typed hex value**, e.g. `#FFFFFF`. No color picker.
Background:	Background color as ARGB hex, e.g. `#00000000` (fully transparent) or `#CC000000` (dimmed scrim).
Alignment:	**Left** / **Center** / **Right** / **Justified**.
Line Spacing:	Multiplier, default 1.0.
Display: / **Display #:**	Output display and stage number.
Mode:	**FillStage** / **Custom** (geometry mode).
Opacity:	0 – 1 slider.
Layer:	0 – 1000.
Blend Mode:	The same 23 blend modes as Video.

A Text cue **runs until stopped** — it has no automatic end. Take it down with a Stop cue, by pressing `S` with it selected, or by replacing it with the next visual cue on that layer.

[Illustration to be added in a future revision]

Worked example: supertitles

1. Press `Ctrl+5`, name it `Title 1 – Act I`.
2. **Text** tab → type the first title line into **Text Content**; **Size:** `64`; **Color:** `#FFFFFF`; **Background:** `#00000000`; **Alignment: Center**; **Display:** the supertitle screen.
3. Important: Text cues **stack** — firing a second title does not remove the first, so each title needs to take the previous one down. The clean pattern: put all the titles in a Group, and on each title's **Triggers** tab set **Fade & Stop Others** to **Peers** with a short fade time (e.g. `0.3`). Now every title, as it fires, fades out and stops its sibling titles — and nothing else in the show.
4. Duplicate the cue (`Ctrl+D`) for each subsequent title and edit the text — duplicates keep the styling and the Fade & Stop Others setting.
5. Fire each title with GO. Add a final **Stop** cue targeting the last title for a clean blackout of the title screen.

Text gotchas

- A typo in **Font:** silently falls back to a default face — type the family name exactly ("Segoe UI", not "Segoe").

- Colors are hex-only: `#RRGGBB` for **Color:**, `#AARRGGBB` with an alpha byte for **Background:**. `#00000000` = invisible background; `#FF000000` = solid black.
- Text cues neither end on their own nor replace each other — two fired Text cues are both on screen. Plan each cue's Stop (or use the Fade & Stop Others pattern from the worked example).

Common issues

- **Audio cue fires but is silent.** Check the **Levels** tab — Main Level may be at `-INF` (use **Set Default Levels**), or the wrong output **Patch:** is set on I/O. Also confirm the file still exists (broken cues fire as silent no-ops).
- **"My Mic cue shows nothing and won't play."** Correct — Mic cues (present only in workspaces from earlier builds; the toolbar button is disabled) are not functional in this release. See the Mic section above.
- **Video plays on the wrong monitor.** The assigned display wasn't found at load, so output fell back to the primary display for this session. Reconnect/re-map the display (**Preferences** → **Displays**) and reload the workspace — your saved assignment was preserved.
- **Video stutters.** Software decoding — re-encode heavy files to H.264 MP4 at your display's resolution, and close other CPU-hungry apps on the show machine.
- **The last video frame froze on screen and won't go away. Hold Last Frame** is checked and holds until stopped — press `S` with the cue selected, fire your next visual cue, or panic.
- **The fade-out seems to start "too early."** By design: fade-out leads into the end so it *completes* at End Time. Set a shorter Fade Out if you want more full-level content.
- **Camera LED is on but nothing is on stage.** Normal — that's the warmup pool holding the device ready. The feed appears when you GO the cue.
- **Camera shows "Not connected."** The device is unplugged or another app owns it. Close the other app, reconnect, and click the refresh button next to **Device:**.
- **I fired the same camera to a second display and the first went dark.** One device drives one output — the feed moves to the newest cue. Use a second camera for simultaneous outputs.
- **My image never goes away.** Its **Hold (s):** is `0` (indefinite). Give it a hold time or stop it explicitly.
- **Text cue shows the wrong font.** The typed family name doesn't match an installed font — check the exact name in Windows' font settings.
- **Two titles are on screen at once.** Text cues stack rather than replace — stop the previous title, or set **Fade & Stop Others** to **Peers** on titles grouped together.

Control & Structure Cues

Not every cue makes sound or light. The fifteen cue types in this chapter organize your cue list, act on *other* cues, pace your show, and drive the Show Timer. Master these and a single press of **GO** can run an entire scene change by itself.

All of them are created the same way as any other cue: click the cue type's button on the toolbar, or press its keyboard shortcut if it has one (see *Working with Cues* for cue creation in general).

Cue	Toolbar button	Shortcut	What it does
Group	**Group**	`Ctrl+0`	Container that plays its children in a chosen mode
Fade	**Fade**	`Ctrl+7`	Fades levels, rate, or geometry of another cue
Start	**Start**	—	Starts (or restarts) another cue
Stop	**Stop**	—	Immediately stops another cue
Pause	**Pause**	—	Pauses another cue
Load	**Load**	—	Pre-loads another cue for instant firing
Reset	**Reset**	—	Stops and resets another cue to its saved state
Devamp	**Devamp**	—	Button disabled — not available in this version (see below)
GoTo	**GoTo**	—	Moves the playhead to another cue without firing it
Target	**Target**	—	Button disabled — not available in this version (see below)
Arm	**Arm**	—	Arms another cue
Disarm	**Disarm**	—	Disarms another cue
Wait	**Wait**	—	Does nothing for its duration — a timed spacer
Memo	**Memo**	—	Does nothing at all — a note for the operator
Timer	**Timer**	—	Controls the Show Timer window

Every type in this table is available on the Free tier. The Timer cue is the cue-list remote for the Show Timer — a Free feature — so it is Free as well.

How control cues pick their target

Group, Wait, Memo, and Timer cues stand alone. Every other cue in this chapter acts on a **target cue**, chosen in the inspector:

1. Select the control cue.
2. Open the **Basics** tab.
3. In the **Target** row, open the drop-down. It lists every cue in the workspace as *number + name*.

4. Pick the cue this control cue should act on.

[Illustration to be added in a future revision]

Things every targeted control cue has in common:

- **They are instant.** Firing one performs its action and completes immediately — there is nothing to watch run.
- **They do not move the playhead** (GoTo is the one deliberate exception). Firing a Stop cue by hotkey mid-show stops its target and leaves your standby position untouched.
- **No target = no action.** A control cue with an empty **Target** is considered broken: it fires, does nothing, and the playhead advances normally. The cue list does not display a broken-cue badge, so an empty Target is easy to miss — check your control cues during tech.
- **Inspector tabs:** the ten transport types (Start through Disarm) have only the **Basics** and **Triggers** tabs. There is nothing type-specific to configure beyond the target.
- **They chain like any cue.** Pre Wait, Post Wait, and the Continue modes all work, so control cues slot naturally into auto-continue sequences.

Group

Toolbar: Group · shortcut `Ctrl+0` · Free tier.

A Group cue is a container. Drag cues into it (or create cues while a child of the group is selected — new cues are inserted inside the group) and the group plays them according to its mode. Group rows show a ▶/▼ toggle in the Name column to collapse or expand their contents.

The fastest way to build one: select several existing cues, right-click, and choose **Group Selected Cues** (`Ctrl+G`). The selection is wrapped in a new Group cue in place.

Inspector tabs: Basics, **Triggers**, **Group**.

The **Group** tab has:

Field	What it does
Mode	How the group plays its children: **List**, **StartFirstEnter**, **StartFirst**, **Timeline**, **StartRandom**, **Cart**, **Playlist**, or **StartAll**
Children	Read-only count of cues inside the group
Playlist Options — **Shuffle**, **Loop**	Shown only in **Playlist** mode

The eight modes range from “fire children one at a time” to “fire everything at once” to the clickable **Cart** grid that replaces the list view. The full mode-by-mode guide, with examples, lives in *Working with Cues* — this chapter won’t duplicate it.

[Illustration to be added in a future revision]

Gotchas

- Mode names appear exactly as written above (for example **StartFirstEnter**, not "Start First and Enter").
- Stopping a group stops its running children.
- In **Cart** mode the group renders as a grid of clickable buttons in the main window — see *Working with Cues* for carts.

Fade

Toolbar: Fade · shortcut `Ctrl+7` · Free tier.

A Fade cue smoothly changes properties of another cue — almost always one that is currently playing. It can fade an audio cue's main level and playback rate, and it can fade a video, image, or camera cue's opacity, position, scale, rotation, and soundtrack volume. Fading geometry works on *moving* video: you can shrink, slide, or dissolve a clip while it continues to play.

Pick the cue to fade in the **Basics** tab's **Target** row, like any control cue.

Inspector tabs: Basics, **Triggers**, **Curve**, **Levels**, **Geometry**.

The Curve tab

Field	Options / units	What it does
Curve shape:	**Linear**, **SCurve**, **Parametric**	The interpolation shape of the fade
Audio level domain:	**Slider**, **Decibel**, **Linear**	Which scale audio levels are interpolated in
Duration (seconds):	seconds	How long the fade takes
Curve intensity:	number (shown only for **Parametric**)	Shapes the parametric curve
Stop target when done	checkbox	Stops the target cue when the fade completes

SCurve eases in and out — the most natural choice for music. **Decibel** domain fades sound like a hand pulling a fader; **Linear** domain fades reach silence more abruptly at the tail.

[Illustration to be added in a future revision]

The Levels tab

Field	What it does
Fade mode:	**Absolute** fades *to* the value you enter; **Relative** fades *by* that amount from wherever the level currently is
Stop target when done	Same checkbox as on the Curve tab — set it on either
Set Levels from Target / **Set All Silent** / **Clear All**	Quick presets: copy the target's current levels, set everything to silence, or clear all fade entries

Field	What it does
Fade destination:	Read-only display of which cue will be faded
Main Level checkbox + **From (current):** + **To (target):**	Enable the main-level fade; **From (current)** shows the level the fade will start from (read live from the playing cue), **To (target)** is where it ends, in dB
Fade rate to	Enable and set a playback-rate fade (for example, slow a track to a stop)
Outputs:	Shows the output count; below it: "Per-output fade targets are coming in a future update."

Per-output fade targets are not part of this release — the placeholder text marks where they will appear. Fade the **Main Level** to control the cue's overall volume.

[Illustration to be added in a future revision]

The Geometry tab

Each row has an enable checkbox and a target value. Enable only the properties you want to fade; everything else is left alone.

Property	Range
Opacity:	0–1 (shown as %)
X Translation: / **Y Translation:**	−5000 to 5000 px
X Scale: / **Y Scale:** (with a 🔒 lock to move them together)	0.01–5.0
Volume %:	0–100
X Rotation: / **Y Rotation:** / **Z Rotation:**	−360 to 360°

The tab has its own **Fade mode:** (**Absolute** / **Relative**) and **Stop target when done** controls.

This is the tool for the classic video dissolve: target a running Video cue, enable **Opacity:** with a target of 0, set the duration on the Curve tab, and tick **Stop target when done**. The video fades out mid-motion and the output closes when the fade lands.

[Illustration to be added in a future revision]

Worked example: fade to silence

End of act — the underscore music needs to disappear gracefully.

1. Cue 18 is your Audio cue, already playing.
2. Create a Fade cue (`Ctrl+7`) after it. Name it "Fade out underscore".
3. **Basics** tab → **Target** → pick cue 18.
4. **Curve** tab → **Curve shape: SCurve**, **Duration (seconds):** 8.
5. **Levels** tab → check **Main Level**, set **To (target):** to `-INF` — or just click **Set All Silent**.

6. Check **Stop target when done**.

On **GO**, the music takes eight seconds to melt away, and the audio cue stops cleanly at the bottom so it isn't left running silently.

[Illustration to be added in a future revision]

Gotchas

- A Fade cue is built to act on a *running* cue. Fire it while the target is playing.
- **Stop target when done** appears on the Curve, Levels, and Geometry tabs — it is one setting, surfaced in each place. Without it, a faded-to-silence audio cue keeps "playing" inaudibly until it ends or is stopped.
- **Relative** mode is handy for "duck everything down 6 dB" moments; **Absolute** is the right choice when the end value matters (silence, full, 50%).

Start

Toolbar: Start · Free tier.

Fires its target cue. If the target is already playing, it restarts from the beginning. The playhead does not move — the audience hears the target start, and your standby cue is still your standby cue.

When a Start cue fires a target, the target's own **Continue** setting still applies, so starting the first cue of an auto-continue chain re-runs the whole chain. That makes Start the building block for loops (see the worked example under GoTo below).

Common uses: firing a sound effect that lives in another cue list, re-triggering walk-in music, kicking off a chain from a hotkey.

Gotcha: a Start cue cannot fire a target that is locked by your license tier — the engine blocks it and logs a warning.

Stop

Toolbar: Stop · Free tier.

Immediately stops its target cue — a hard cut, with no fade. Video output for the target closes; audio stops on the spot.

Common uses: the end-of-scene blackout button — put a Stop cue (or several, one per running element) at the top of the next scene so one **GO** silences whatever was left running; killing a looping effect.

> **Why it works this way:** Stop is deliberately instant so it always means the same thing under pressure. When you want a *graceful* stop, that's a Fade cue with **Stop target when done** — the two cues split the job of "stop it now" versus "take it out nicely."

Pause

Toolbar: Pause · Free tier.

Pauses its target cue in place. Audio holds its position; a paused cue shows a pause badge and a yellow left border in the cue list.

A Pause cue only pauses — it is not a toggle. Resuming the cue is a manual operation from the transport controls (see *Working with Cues* for pausing and resuming playback).

Common use: holding an underscore during an unpredictable audience moment, ready to resume on cue.

Load

Toolbar: Load · Free tier.

Pre-loads its target cue so it is ready for a zero-latency start: the target is returned to its ready state and its media (audio or video) is loaded ahead of time.

Common use: just before a musical number that must land exactly on a visual, load the track a few cues early so the **GO** is instantaneous.

A Load cue lets you schedule the load as an explicit step in your sequence. (A per-cue automatic-load option is planned for a future update.)

Reset

Toolbar: Reset · Free tier.

Stops its target if it is running or paused, and resets it to its saved state — elapsed times cleared, ready to fire fresh from the top.

Common use: a "rewind the deck" cue at the end of rehearsal sections, so jumping back and re-running a sequence always starts from a clean state.

Devamp

Toolbar: Devamp · Free tier.

The Devamp cue is not available in this version. Its toolbar button is visible but disabled. Devamp cues loaded from a workspace saved by an earlier build still appear in the list, but firing one has no effect on its target.

GoTo

Toolbar: GoTo · Free tier.

Moves the playhead to its target cue *without firing it*. If the target lives in a different cue list, ATTACCA switches to that list too. The next **GO** fires the target.

Common uses: jumping the playhead back to the top of preshow after a rehearsal run; skipping over an optional section; “choose your own adventure” branching driven by hotkeys.

Worked example: preshow loop

You want walk-in music that loops seamlessly until the show starts, and you want the playhead parked ready for the top of the show.

Cue	Type	Settings
1 “Walk-in music”	Audio	your track; **Continue** = **AutoFollow**
2 “Loop walk-in”	Start	**Target** = cue 1
3 “House out — top of show”	(your opening sequence)	—

Press **GO** on cue 1. When the track ends, **AutoFollow** fires cue 2, which restarts cue 1 — a seamless loop. Because a Start cue never moves the playhead, the loop spins in the background while you sit with the playhead on cue 3, ready for the show.

To end the loop, stop the audio cue — from the transport, or with a Stop cue targeting cue 1 built into your opening sequence. A stopped cue does not trigger its **AutoFollow**, so the loop breaks cleanly.

Prefer an operator-paced loop instead? Replace the Start cue with a GoTo cue targeting cue 1 and set the audio cue’s **Continue** to **DoNotContinue**: each time the track ends, the playhead is already back on cue 1 and you press **GO** whenever you want it again.

[Illustration to be added in a future revision]

Target

Toolbar: Target · Free tier.

The Target cue is not available in this version. Its toolbar button is visible but disabled. Target cues loaded from a workspace saved by an earlier build still appear in the list, but the retargeting fields are not exposed in the inspector, so firing one changes nothing.

Arm

Toolbar: Arm · Free tier.

Sets its target cue’s **Armed** checkbox on. An armed cue fires normally from **GO** and from its triggers.

Disarm

Toolbar: Disarm · Free tier.

Sets its target cue’s **Armed** checkbox off. Disarmed cues render dimmed in the cue list and are skipped — they won’t fire.

Arm/Disarm pairs let the cue list reconfigure itself mid-show: disarm the pyro cue on the night the effect is cut, arm the alternate ending, disable a block of cues during the matinee. Note that this is the per-cue **Armed** flag — it is separate from the master arming of time-based triggers described in *Working with Cues*.

Wait

Toolbar: Wait · Free tier.

A Wait cue does exactly nothing — for a precise amount of time. Its **Duration** (Basics tab, in `H:MM:SS.ff`, `M:SS.ff`, or plain seconds) *is* the wait. It exists to put deliberate, visible time between steps of an automated sequence: fire it, and the next cue's **AutoFollow** or the wait's own completion paces what happens next.

Inspector tabs: Basics, **Triggers** only.

How is this different from Pre Wait? Every cue has a **Pre Wait** field that delays its own action — that's invisible plumbing inside one cue. A Wait cue is a *row in the list*: the operator sees it counting down in the **Duration** column, it can be stopped or panicked like anything else, and it can carry its own triggers. Use Pre Wait for a small private offset; use a Wait cue when the gap is a real beat of the show that deserves its own line.

[Illustration to be added in a future revision]

Worked example: automated scene change

One **GO** at the end of Scene 3 should fade the music, hold ten seconds of transition, then bring in the next scene's soundscape.

Cue	Type	Settings
30 "Fade out Scene 3 music"	Fade	Target = the running music cue; Duration 3 s; **Set All Silent**; **Stop target when done**; **Continue** = **AutoContinue**
31 "Scene change"	Wait	**Duration** = 10 s; **Continue** = **AutoFollow**
32 "Scene 4 soundscape"	Audio	your next track

Press **GO** once. **AutoContinue** on cue 30 fires the Wait immediately, so the ten-second clock runs *while* the three-second fade finishes. When the Wait completes, **AutoFollow** fires cue 32. If the scene change needs a hard cut instead of a fade, swap cue 30 for a Stop cue — everything else stays the same.

[Illustration to be added in a future revision]

Memo

Toolbar: Memo · Free tier.

A Memo cue performs no action at all — it completes instantly. It is a note that lives in the running order: "CHECK: follow spot ready", "Intermission — house to 50%", "SM calls fly cue 12 here". Its name and its **Notes** field (Basics tab) are its whole content, and the notes appear in the header notes area when the cue is selected.

Inspector tabs: Basics, **Triggers** only.

Because it honors the Continue modes like any cue, a Memo placed inside an auto-continue chain passes straight through without disturbing the timing. Give memos a strong **Color** on the Basics tab so they read at a glance in a dark booth.

Timer

Toolbar: Timer (tooltip "Timer Cue — control Show Timer window").

A Timer cue drives the Show Timer — the big clock/countdown/message window you put on a backstage monitor. The window itself (opening it, `Ctrl+T`, fullscreen, mirrors, appearance) is covered in *Running the Show*; the Timer *cue* is how the cue list controls it, so your countdowns and cast messages fire on **GO** along with everything else. Timer cues complete instantly and chain cleanly with auto-continue.

Inspector tabs: Basics, **Triggers**, **Timer**.

The **Timer** tab:

Field	Shown for	What it does
Action	always	What this cue does to the Show Timer when fired — one of the eight actions below
Duration	**StartCountdown** only	Countdown length, entered as `MM:SS`, `HH:MM:SS`, or seconds (default 60 s)
Label / Message	text-relevant actions	Text displayed under the countdown, or the message body
Display Sec.	**SetMessage** only	How long the message stays up; 0 = stays until cleared by the next Timer cue
Override font size	**SetMessage**	Per-message font size (default 48)
Override font color	**SetMessage**	Per-message color, `#AARRGGBB` hex
Override background color	**SetMessage**	Per-message background, `#AARRGGBB` hex

The eight actions, exactly as they appear in the **Action** drop-down:

Action	Effect
StartCountdown	Starts a countdown of **Duration**, with the optional **Label / Message** under it
StopCountdown	Stops and clears the countdown
PauseCountdown	Freezes the countdown
ResumeCountdown	Resumes a paused countdown
StartShowTimer	Starts the elapsed show timer (counts up)
StopShowTimer	Stops the elapsed show timer

Action	Effect
ResetShowTimer	Resets the elapsed show timer to zero
SetMessage	Displays a message in the operator message bar, with optional timing and style overrides

[Illustration to be added in a future revision]

A typical top-of-show pattern: cue 0.5 “Places — 5 minutes” is a Timer cue with **Action** = **StartCountdown**, **Duration** `5:00`, **Label / Message** “PLACES”; the house-open sequence ends with a Timer cue whose **Action** = **StartShowTimer** so elapsed time starts ticking the moment the show truly begins; and a **SetMessage** cue mid-act can flash “QUIET BACKSTAGE” to every timer display in the building.

[Illustration to be added in a future revision]

Common issues

- **My Stop/Start/Fade cue fires but nothing happens.** Its **Target** (Basics tab) is probably empty, or the cue it pointed at was deleted. A targetless control cue fires silently and does nothing, and the cue list shows no broken badge — re-pick the target.
- **The music faded out but the cue still shows as running.** The Fade cue reached silence but **Stop target when done** wasn’t checked. The target keeps playing inaudibly to its natural end. Check the box (it’s on the Curve, Levels, and Geometry tabs — one setting).
- **My loop won’t stop.** If you built a Start-cue loop, stop the *audio* cue (transport stop, or a Stop cue targeting it). Stopping only the Start cue does nothing — it completed instantly the moment it fired.
- **GoTo fired the cue it points at.** It doesn’t — GoTo only moves the playhead. If the target played, something else fired it: most likely you pressed **GO** again, or the previous cue’s **Continue** mode chained into it.
- **The playhead jumped to a different cue list.** A GoTo cue targeting a cue in another list also switches to that list. That’s by design — retarget the GoTo if it’s pointing at the wrong copy of a cue.
- **My Devamp / Target cue does nothing.** Correct — this applies to cues loaded from workspaces saved by earlier builds (both types’ toolbar buttons are disabled in this release, so new ones can’t be created). Devamp awaits slice-based playback; the Target cue’s retargeting fields are not yet in the inspector.
- **A Pause cue paused my cue, but nothing resumes it.** By design, Pause is one-way. Resume from the transport controls, or restart the cue with a Start cue (which restarts from the beginning, not from the pause point).
- **I disarmed a cue with a Disarm cue, and now it’s skipped every night.** Armed state is part of the workspace and persists. Pair every Disarm with an Arm somewhere (or re-check **Armed** on the Basics tab) when the change was meant to be temporary.

Show Control Cues

This chapter covers the ten cue types that talk to the world outside ATTACCA: **Light**, **Effect**, **Network**, **MIDI**, **MIDI File**, **Timecode**, **Script**, **Serial**, **HTTP**, and **OSC**. Between them they can drive lighting rigs, media servers, mixing consoles, keyboards, projectors, relays, web APIs, and anything else that speaks DMX, MIDI, timecode, RS-232, HTTP, or OSC.

Every cue type in this chapter requires ATTACCA Pro. On the Free tier, clicking any of these toolbar buttons opens the Pro Feature dialog instead of creating a cue, and these cue types in a loaded workspace show a gray lock glyph in the cue list (tooltip: **This cue requires ATTACCA Pro**) and cannot be fired or edited. This is stated once here rather than repeated in every section below.

All ten are created from the toolbar (there is no Cue menu — cue creation is toolbar buttons and keyboard shortcuts only). Two have default create-shortcuts: Light (`Ctrl+6`) and MIDI (`Ctrl+9`). The rest are toolbar-only by default, though you can assign a shortcut to any of them in **Preferences** → **Keyboard Shortcuts**. One exception: the **Network** button is visible but disabled — the Network cue is not available in this version; see the Network section below.

Every cue in this chapter gets the shared **Basics** and **Triggers** inspector tabs, covered earlier in this manual; the sections below describe only each type's own tabs. One note on dropdowns before you start: several fields in these inspectors display raw internal option names such as `Fps2997NonDrop` or `SacnMulticast` . This manual quotes those options exactly as the app shows them.

Light cues

A Light cue sets DMX levels — a lighting "look." When it fires, the channels it addresses fade to their programmed values over the cue's fade time, and they *stay* there after the cue completes (this is standard behavior in professional lighting — levels hold until another cue changes them). Light cues output over sACN or Art-Net, configured in **Preferences** → **Lighting** (see *Lighting* for network setup and fixture patching).

- **Toolbar button: Light** — shortcut `Ctrl+6`
- **Inspector tabs: Basics**, **Triggers**, **Levels**, **Curve**, **Recording**
- **Targeting:** none — a Light cue doesn't target another cue or a file; its content is the levels you program on the Levels tab.

The Levels tab

This is where you build the look. It is a large tab with several sections, top to bottom.

[Illustration to be added in a future revision]

DMX Channel Grid. A **Universe:** dropdown selects which universe the grid displays. The grid itself is 16 columns of channel cells; each cell shows the channel number, an editable DMX value (0–255), and — when a fixture is patched at that address — the parameter label from the fixture profile (e.g. Dimmer, Red, Pan). Type

directly into a cell to set that channel.

Select Range. For bulk programming: enter **Start:**, **End:**, and **Value:**, then click **Set All** to write one value across a whole channel range.

Channel Faders (used channels). Every channel that has a value in this cue also appears as a vertical fader — drag or click to set levels by feel instead of by number.

Preset Colors (for LED fixtures). Seven one-click swatch buttons: **Red**, **Green**, **Blue**, **White**, **Amber**, **Warm**, **Cool**. These write sensible values to the color channels of LED fixtures.

Color Picker. The heart of the tab for color work:

- **LIVE EDIT** checkbox — when on, every change you make goes to the DMX output *immediately* (a ● indicator with the text **(live to wire)** confirms it); when off, changes are stored in the cue and only heard when it fires. Live Edit is how you program while looking at the actual rig.
- **Group:** dropdown plus **All** / **None** buttons and a per-fixture checklist — choose which patched fixtures receive the color. Each row shows the fixture's name, address, and capability, with a selection summary underneath.
- With no fixtures selected, manual mode appears instead: **Layout:** (`RGB` , `RGBW` , `RGBA` , `CMY`), **Start ch:** (the first color channel), and a **White mix:** slider for layouts with a white emitter.
- **Intensity:** slider (0–100%) dims the selected fixtures.
- **Pan:** and **Tilt:** sliders (0–255, centre = 128) appear when the selected fixtures have movement channels; each has a ↺ reset button.
- A full color wheel for freehand picking.
- **Quick swatches** — fourteen one-click colors: Warm White, Cool White, Red, Green, Blue, Amber, Cyan, Magenta, Purple, Orange, Deep Blue, Lavender, Congo Blue, and No Color / Blackout.
- **My presets** with a **Save current…** button — saves your current color as a named preset via the **Save Color Preset** dialog (prompt: **Preset name:**). Right-click a saved preset and choose **Delete** to remove it.
- **Gels (Rosco / Lee)** — an expandable gel library picker. Filter by manufacturer, search by name or number, and click **Apply** on a gel to use it.

[Illustration to be added in a future revision]

[Illustration to be added in a future revision]

Utility controls. Blackout All zeroes every channel in every universe of this cue. **View mode** switches the levels display between `Slider` , `Tile` , and `Text` ; **Show as %** displays 0–100% instead of raw 0–255 DMX values.

Behavior checkboxes. Collate effects of previous light cues controls whether this cue's look combines with earlier light cues (channels this cue doesn't address hold their current values). **Use as subcontroller in dashboard** exposes this cue as a fader in the Light Dashboard.

Light commands. A free-text command box — one command per line, e.g. `ch1 = 100` or `backlight.zoom = 50` . Invalid lines show a red validation error under the box. Command syntax is covered in full in *Lighting*.

The Curve tab

Controls how the cue fades: **Curve type:** (`Custom` , `Linear` , `SCurve` , `Parametric`), **Duration (seconds):** for the fade time, a **Parametric intensity:** field when the Parametric curve is selected, and **Use split fade times (separate up/down)** which reveals separate **Up fade:** and **Down fade:** fields. There is no graphical curve editor — the dropdown is the whole control. Fade behavior and split-fade conventions are covered in *Lighting*.

The Recording tab

Lets a Light cue capture live DMX from an external console (over `SacnMulticast` or `ArtNet`) and play the recorded stream back instead of static levels — complete with a ▶ **Check Signal** input monitor, a ● **Record** / ■ **Stop** / ✕ **Clear** transport, a draggable trim/loop timeline, a **Loop mode:** selector (Off / Loop as recorded / Auto-trim loop), a per-cue crossfade, per-cue **Fade in** / **Fade out** playback times in seconds (default 0), and an **On finish** setting for non-looping takes (Hold last frame — the default — / Blackout / Go to cue → any other Light cue, fired as a normal GO). The cue list's Duration column shows the take's length (∞ when looping), and the Curve tab is greyed out while a recording is attached — recorded playback has no fixture fade. A read-only **Playback output:** line shows what this computer transmits, as set in Preferences → Lighting. Takes are saved in a `recordings` folder next to the workspace so they travel with the show. Firing a recorded cue silently takes over any universes another recorded cue is using — no flicker, no blackout. Recording is a substantial workflow of its own; see *Lighting* for the full capture, loop, and Park-cue guide.

> *[Illustration to be added in a future revision]*

Worked example: a warm stage wash

1. Patch your fixtures first (see *Lighting*), then click **Light** in the toolbar (or press `Ctrl+6`).
2. On the **Levels** tab, check **LIVE EDIT** so you can see the rig respond as you work.
3. In the **Color Picker** section, click **All** to select every patched fixture, then click the **Warm White** quick swatch.
4. Drag **Intensity:** to about 80%.
5. On the **Curve** tab, set **Duration (seconds):** to `3` for a gentle 3-second fade.
6. Uncheck **LIVE EDIT**, press `Esc` to clear the rig if needed, then fire the cue with **GO** to confirm the fade.

Gotchas

- With **LIVE EDIT** off, nothing you program is heard until the cue fires. With it on, everything is — don't program live during a performance.
- Channels hold their last value after a cue completes. To go dark, fire a cue that explicitly sets channels to 0 (or use **Blackout All** in a dedicated blackout cue).
- The fade time lives on the **Curve** tab's **Duration (seconds):** field, not the Basics tab.
- If nothing reaches the rig at all, check that sACN or Art-Net output is enabled in **Preferences → Lighting** — see *Lighting*.

Effect cues

An Effect cue runs a repeating pattern — a chase, a sine wave, a color cycle — *on top of* the current lighting state. Where a Light cue sets levels once, an Effect cue keeps modulating them for as long as it runs. Use it for pulsing pars, police chases, rainbow sweeps, and strobes without programming dozens of individual cues.

- **Toolbar button: Effect** — no default shortcut (tooltip: *Effect Cue — chase / sine / colour cycle running on top of cue state*)
- **Inspector tabs: Basics**, **Triggers**, **Effect**
- **Targeting:** the effect targets a fixture group (or all patched fixtures) via the **Fixtures:** field — not another cue.

The Effect tab

[Illustration to be added in a future revision]

Preview. A live waveform at the top of the tab draws one cycle of the configured pattern — the caption reads *One cycle of the configured pattern. Changes as you edit.* It updates as you adjust any parameter.

Core fields:

Field	Options / range	What it does
Type:	`Chase`, `Sine`, `Ramp`, `Pulse`, `Strobe`, `Random`, `ColorCycle`, `Rainbow`, `ColorChase`	The pattern this cue runs
Target:	`Intensity`, `Color`, `Pan`, `Tilt`	Which fixture parameter the effect modulates
Fixtures:	`All Patched Fixtures`, plus your fixture groups	Which instruments the effect drives; empty = every patched instrument
Presets:	pattern-specific buttons	Click a preset to populate every parameter below — you can still tweak after
Waveform Shape:	`Sine`, `Ramp`, `Triangle`, `Square`	Wave shape (Sine, Ramp, and Strobe types pick their own shape and ignore this)
Direction:	`Forward`, `Reverse`, `Bounce`, `Random`	Travel direction across the fixture group
Width / Duty:	0–100% slider	The "on" fraction of each cycle
Phase Spread:	0–360° slider	Per-fixture phase offset — what turns a pulse into a chase
Distribution:	`Spread`, `None`, `Random`, `Pairs`, `OddsEvens`	How the phase offset is distributed across fixtures
Amplitude:	0–200% slider	Modulation depth; 100% = full ±127 DMX

Field	Options / range	What it does
Rate (Hz):	number	Speed — 1.0 Hz is one full cycle every second
BPM: + **Tap Tempo** + **Reset**	BPM = Rate × 60	Tap in rhythm (2+ taps within 2 s sets BPM to the average interval); **Reset** clears tap history
Fade In (s): / **Fade Out (s):**	seconds	Ramps the effect's amplitude in and out

Colors section (appears for color effects): a **Mode:** choice of **Chase** / **Smooth Fade** / **Rainbow** radio buttons, a swatch grid with **+ Add Color** and per-swatch **×** remove buttons (click a swatch to open a color-picker popup), and a **Sequence preview:** gradient strip showing what will be emitted. The swatch list is hidden in Rainbow mode, which generates its own spectrum.

[Illustration to be added in a future revision]

About presets: the **Presets:** buttons on this tab are built-in quick-starts, one set per pattern type — they fill in the parameters below and you tweak from there. They are not user-saveable. (The **Save Effect Preset** dialogs you may have seen elsewhere belong to *audio* effect chains on Audio cues, and the **Paste Effects** dialog likewise handles copied audio effect chains — neither is related to Effect cues.)

Worked example: a four-fixture intensity chase

1. Click **Effect** in the toolbar.
2. Set **Type:** to `Chase` and **Target:** to `Intensity`.
3. Set **Fixtures:** to your front-wash group (or leave it on `All Patched Fixtures`).
4. Set **Rate (Hz):** to `2` — or click **Tap Tempo** in time with the music.
5. Set **Phase Spread:** to 360° and **Distribution:** to `Spread` so the four fixtures fire in sequence rather than together.
6. Watch the preview waveform, then fire the cue. Stop it like any other cue when the number is over.

Gotchas

- An Effect cue modulates whatever levels are already on stage — if the fixtures are at zero and the target is `Intensity` with a modest amplitude, you may see very little. Bring a base look up with a Light cue first, or push **Amplitude:** higher.
- **Fixtures:** left empty means *every* patched instrument. That's a feature until it isn't.
- `Sine`, `Ramp`, and `Strobe` types ignore the **Waveform Shape:** field — they use their own shape.
- Panic stops effects with the same escalation as everything else: single `Esc` fades them out over the panic duration, double `Esc` within 1 second stops them instantly.

Network cues

A Network cue sends a network message — an OSC message, a line of plain text, or raw hex bytes — to a predefined network output patch. It also supports *fading* a value inside the message over the cue's duration, resending the message many times per second with an interpolated number substituted in. That makes it the tool for continuously animating a parameter on a media server or console, where the OSC cue (below) sends exactly one message.

- **Toolbar button: Network — disabled; the Network cue is not available in this version**; no default shortcut (`Ctrl+8` is intentionally unassigned)
- **Inspector tabs: Basics**, **Triggers**, **Settings**
- **Targeting:** the destination host, port, and protocol come from the workspace's network output patch identified by the **Patch** number — not from fields on the cue.

The Settings tab

[Illustration to be added in a future revision]

Field	Options / range	What it does
Patch	number; 0 = unpatched	Which network output patch to send through
Type	`Osc` , `PlainText` , `Hex`	Message format
Fade Mode	`NoFade` , `Resend` , `OneDFade` , `TwoDFade`	Whether and how the message repeats/interpolates over the cue's duration
Message Content	multiline text	The message body — *OSC address + arguments, plain text, or hex bytes*. Use the `#v#` token for 1D fades, `#x#` / `#y#` for 2D fades
FPS	1–120 (visible only when fading)	Messages per second while fading
1D Fade Values — From / To / Value Type	numbers; `Integer` or `Float` (visible only for `OneDFade`)	The range interpolated into the `#v#` token

For `TwoDFade` there are no dedicated From/To fields — the `#x#` and `#y#` tokens are documented only in the Message Content tooltip.

Worked example: fading a media-server layer opacity

The idea: a `OneDFade` Network cue with **Message Content** of `/composition/layers/1/video/opacity #v#` , **1D Fade Values** From `0` To `1` , **Value Type** `Float` , **FPS** `30` , and a 5-second **Duration** on the Basics tab would ramp a Resolume layer's opacity from 0 to 1 over five seconds — thirty interpolated messages per second. Read the gotcha below before building this.

Gotchas

- **In this release the Network cue cannot be created — its toolbar button is disabled.** The reason: network output patches cannot be created or edited anywhere in the app (the **Patch** number refers to a patch list stored in the workspace, and no window or preferences page exists yet to populate it), so a fired Network cue would find no matching patch, write a warning to the log, and send nothing. Network cues loaded from a workspace saved by an earlier build still appear and their Settings tab is editable, but they remain inert. Use the **OSC** cue (which has its own Host/Port fields and a Test Send) for one-shot OSC, and the **HTTP** or **Serial** cues for other transports. We document the Network cue's fields here so the inspector isn't a mystery for legacy workspaces.
- The fade tokens (`#v#` , `#x#` , `#y#`) are literal text in **Message Content** — the cue substitutes numbers into them at send time.

MIDI cues

A MIDI cue sends a single MIDI message when it fires: a musical "voice" message (note, control change, program change…), an MSC (MIDI Show Control) command, or a raw System Exclusive packet. It's how ATTACCA talks to keyboards, sound consoles, other show-control systems, and anything else with a MIDI input.

- **Toolbar button: MIDI** — shortcut `Ctrl+9`
- **Inspector tabs: Basics**, **Triggers**, **Settings**
- **Targeting:** the **Destination** dropdown selects a MIDI output patch. Patches are created automatically — each MIDI output device you enable in **Preferences** → **MIDI** becomes a patch. No devices enabled means nothing to send to.

The Settings tab

The **Type** dropdown (`Voice` , `Msc` , `SystemExclusive`) selects which of three sections is shown, and **Destination** picks the output patch.

[Illustration to be added in a future revision]

Voice Message section (Type = `Voice`):

Field	Options / range	What it does
Learn	button (shows **Listening…** while active)	Captures the next incoming MIDI message and fills in the fields below — enabled only when a MIDI input device is configured
Command	`NoteOff` , `NoteOn` , `PolyTouch` , `ControlChange` , `ProgramChange` , `ChannelPressure` , `PitchBend`	The voice message type
Channel	1–16	MIDI channel

Field	Options / range	What it does
(first data byte — label changes with Command: **Note** / **Controller** / **Program** / **Pressure** / **Value** / **Byte 1**)	0–127; note names or numbers via the **Note/Number** toggle	The message's first data byte. PitchBend instead shows a single **Value** field (0–16383)
(second data byte — label changes: **Velocity** / **Value** / **Pressure** / **Byte 2**)	0–127	The second data byte; hidden for ProgramChange and ChannelPressure, which don't have one
Fade + **Start** / **End**	fadeable messages only	Sweeps the value from Start to End over the cue's duration — e.g. a CC ramp

MIDI Show Control section (Type = `Msc`): **Command** (the MSC command byte — 1=GO, 2=STOP, 3=RESUME, 4=LOAD), **Format** (command format byte; 127 = All Types), **Device ID** (0–127; 127 = all-call), **Cue #**, **Cue List**, and **Cue Path**. This is MSC *output* — firing this cue tells another MSC device to GO. (ATTACCA can also *respond* to incoming MSC; that lives in **Preferences → MIDI** and is covered in *Control & Integration*.)

[Illustration to be added in a future revision]

System Exclusive section (Type = `SystemExclusive`): a multiline **Hex data (F0/F7 framing optional):** box. Enter hex bytes separated by spaces, e.g. `F0 43 10 3E ... F7`.

Worked example: a program change to a keyboard rig

Your keyboard player's rig switches sounds on MIDI program changes. To switch it to patch 12 at the top of a song:

1. Enable your MIDI interface's output in **Preferences → MIDI** (check the master **Enable MIDI output (send)** box and the device's own checkbox). The device automatically becomes a patch.
2. Click **MIDI** in the toolbar (or press `Ctrl+9`) and name the cue on the Basics tab — e.g. "Keys → Patch 12".
3. On the **Settings** tab: **Type** `Voice`, **Destination** = your interface, **Command** `ProgramChange`, **Channel** `1`, **Program** `11`.
4. Fire the cue with **GO** and confirm the rig switches. Then set the cue's **Continue** mode so it rides along with the adjacent audio or lighting cue.

> **Why it works this way:** many devices display programs 1–128 but transmit 0–127 on the wire. If the rig lands one patch off, that's the off-by-one — enter the displayed number minus one.

Tip — driving software on the same computer: MIDI normally connects two devices, but a free virtual MIDI driver such as loopMIDI (Tobias Erichsen) creates a loopback port that ATTACCA can send into and another application — Resolume Arena, a DAW, a soft-synth host — can listen to. Install loopMIDI, create a port, enable that port as a MIDI output in **Preferences → MIDI**, pick it as the cue's **Destination**, and select the same port as the input inside the receiving app. (loopMIDI carries MIDI only — it does not route audio.)

Gotchas

- There is no Test button on the MIDI cue — fire the cue with **GO** to test it (Edit Mode plays cues normally).
- If the **Destination** dropdown is empty, no MIDI output device is enabled in **Preferences** → **MIDI**. If a cue fires with no valid destination, ATTACCA logs a warning and sends nothing.
- **Learn** is only enabled when a MIDI *input* device is enabled — it captures from your controller to fill the fields, which is handy but not required.
- **Fade** only appears for message types where sweeping a value makes sense; ProgramChange, for instance, can't fade.

MIDI File cues

A MIDI File cue plays an entire Standard MIDI File (`.mid` / `.midi`) out of a MIDI output patch — every track, every event, in time. Use it for pre-sequenced keyboard parts, click tracks to a MIDI metronome, or any longer MIDI performance you'd rather not rebuild as individual cues.

- **Toolbar button: MIDI File** — no default shortcut
- **Inspector tabs: Basics**, **Triggers**, **Settings**
- **Targeting:** this is a file-target cue — the Basics tab shows the file path read-only, and the **Browse…** button lives on the Settings tab.

The Settings tab

[Illustration to be added in a future revision]

Field	Options / range	What it does
Patch	number ≥ 1	The MIDI output patch to play through (same auto-created patches as the MIDI cue)
MIDI File + **Browse…**	file picker — dialog title **Select MIDI File**, filter `MIDI Files (*.mid;*.midi)`	Sets the file target
Rate	multiplier (1.0 = normal speed)	Playback rate

Worked example

1. Click **MIDI File** in the toolbar.
2. On the **Settings** tab, click **Browse…** and pick your sequence.
3. Set **Patch** to `1` (the default patch — your first enabled output device).
4. Fire with **GO**; stop it like any running cue.

Gotchas

- The same prerequisite as MIDI cues: an output device must be enabled in **Preferences** → **MIDI**, or there is nothing behind the patch number.
- **Rate** rescales the whole performance — 0.5 is half speed, 2.0 is double.

Timecode cues

A Timecode cue *generates* timecode: MTC (MIDI Timecode) out of a MIDI patch, or LTC (Linear Timecode, the classic audio signal) out of an audio output. Fire it and ATTACCA becomes the timecode master that other devices — lighting consoles, media servers, broadcast gear — chase.

- **Toolbar button: Timecode** — no default shortcut
- **Inspector tabs: Basics**, **Triggers**, **Settings**
- **Targeting:** none — output routing is chosen on the Settings tab.

(The reverse — ATTACCA *chasing* incoming timecode and firing cues at timecode positions — is **not available in this version**: the Timecode trigger and the Preferences → Timecode input settings are present but inactive; see *Control & Integration*.)

The Settings tab

[Illustration to be added in a future revision]

Field	Options / range	What it does
Type	`Mtc`, `Ltc`	MIDI Timecode vs. audio Linear Timecode
MIDI Patch	patch list (visible only for `Mtc`)	Where MTC quarter-frames are sent
Audio Patch	patch list (visible only for `Ltc`)	The audio destination for the LTC signal
Cue Output	channel number (visible only for `Ltc`)	Which audio output channel carries the LTC (0 is invalid)
Framerate	`Fps23976`, `Fps24`, `Fps24975`, `Fps25`, `Fps2997`, `Fps2997NonDrop`, `Fps30Drop`, `Fps30NonDrop`	SMPTE format — shown with these exact names
Start Time	`HH:MM:SS:FF`	The first transmitted frame
End Time	`HH:MM:SS:FF`, or empty	The final frame; empty = run until stopped

Worked example: striping MTC to a lighting console

1. Enable your MIDI interface's output in **Preferences** → **MIDI** and connect it to the console's MIDI in.

2. Click **Timecode** in the toolbar. On **Settings**: **Type** `Mtc`, **MIDI Patch** = your interface, **Framerate** `Fps30NonDrop` (match the console!), **Start Time** `01:00:00:00`, **End Time** empty.
3. Set the console to chase MTC at the same frame rate.
4. Fire the cue at the top of the show; stop it at the end. Every device chasing the code follows.

Gotchas

- Frame-rate mismatches are the classic timecode failure: both ends must agree, including drop vs. non-drop. The framerate names shown above are exactly what the dropdown displays.
- An empty **End Time** means the generator free-runs until you stop the cue — usually what you want for a show master.
- LTC needs a real audio path: a valid **Audio Patch** and a nonzero **Cue Output** channel, physically connected to the chasing device's timecode input.

Script cues

A Script cue runs a Lua script inside ATTACCA. Scripts can fire, stop, and modify other cues, wait, log, and glue together logic that no single cue type covers — "if the walk-in music is still playing, fade it and then GO." Each run executes in a fresh sandbox with a hard timeout, so a runaway script can't take the show down with it.

- **Toolbar button: Script** — no default shortcut
- **Inspector tabs: Basics**, **Triggers**, **Script**
- **Targeting:** none — scripts reach other cues through the Lua API (`cue("number")` and friends).

The Script tab

[Illustration to be added in a future revision]

Control	What it does
Run	Executes the script immediately, without firing the cue (disabled while a run is in progress)
Validate	Syntax-checks the script without running it
Clear Output	Empties the output console
Timeout: … s	Per-cue execution limit in seconds — default 5, range 0.1–600
(editor)	The script itself — a monospace editor that accepts tabs
(error bar)	A red bar appears under the editor with any validation or runtime error
Output:	A read-only console showing everything the script logs or prints

A taste of the API — this finds cue 1, fires it, and logs:

```lua
local c = cue("1")
if c then
    c:go()
    log("Fired cue " .. c:get_number())
end
```

The full Lua API — cue lookup and control methods, property getters/setters, `wait()`, `workspace.go()` / `stop_all()` / `panic()`, and the sandbox rules — is documented in *Control & Integration*. This section won't duplicate it.

Gotchas

- Scripts that exceed their **Timeout:** are terminated with *Script timed out (exceeded Ns limit)*. `wait()` counts toward the limit — a script that waits 10 seconds needs a timeout longer than 10 seconds.
- The sandbox deliberately removes file, OS, and module access (`io`, `os`, `require`, and similar). Standard string, math, and table libraries are available.
- **Run** executes the script right now, in Edit Mode, against your real cues — a `panic()` in a test run is a real panic.

Serial cues

A Serial cue transmits a command over an RS-232 serial (COM) port. Projectors, video switchers, motorized screens, and plenty of older AV gear are controlled this way. The cue sends its payload as ASCII text or raw hex bytes, with optional line terminators.

- **Toolbar button: Serial** — no default shortcut (tooltip: *Serial/RS-232 Cue*)
- **Inspector tabs: Basics**, **Triggers**, **Settings**
- **Targeting:** none — the destination is the COM port configured on the Settings tab.

The Settings tab

[Illustration to be added in a future revision]

Field	Options / range	What it does
Port	editable dropdown of detected COM ports (e.g. `COM3`)	The serial port to transmit on — pick from the list or type a name
Refresh	button	Re-scans for available serial ports
Baud Rate	`9600`, `19200`, `38400`, `57600`, `115200`	Line speed
Data Bits	5–8	Data bits per frame
Parity	0–4 (0=None, 1=Odd, 2=Even, 3=Mark, 4=Space)	Parity — entered as the numeric code

Field	Options / range	What it does
Stop Bits	0–3 (0=None, 1=One, 2=Two, 3=OnePointFive)	Stop bits — also a numeric code
Command	multiline text	The payload — ASCII text or hex bytes
Encoding:	**ASCII** / **Hex** radio buttons	How the Command text is interpreted
Append CR / **Append LF**	checkboxes	Adds carriage return / line feed after the payload
Test Send	button + status text	Transmits the command to the configured port right now

Worked example: powering on a projector

Your projector's manual says: 9600 baud, 8 data bits, no parity, 1 stop bit; power-on command `PWR ON` followed by carriage return.

1. Connect the projector's RS-232 port (USB-serial adapters are fine) and click **Serial** in the toolbar.
2. **Port** = the adapter's COM port (click **Refresh** if it isn't listed), **Baud Rate** `9600`, **Data Bits** `8`, **Parity** `0`, **Stop Bits** `1`.
3. **Command**: `PWR ON` — **Encoding: ASCII**, and check **Append CR**.
4. Click **Test Send**. The status text confirms the transmit; the projector should start warming up.
5. Place the cue at the top of your preshow sequence with an **Auto-Continue**.

Gotchas

- Parity and stop bits are entered as *numeric codes* (shown in the table above), not names — a quirk worth double-checking against the device manual.
- Only one application can hold a COM port at a time. If **Test Send** errors, close whatever terminal program you were testing the device with.
- Most serial devices require a terminator — if the device ignores you, the missing **Append CR** checkbox is the first suspect.

HTTP cues

An HTTP cue makes a web request when it fires — GET, POST, PUT, or DELETE, with custom headers and a body. Smart relays, PDUs, REST-controlled media servers, streaming encoders, and home-automation bridges all become cueable.

- **Toolbar button: HTTP** — no default shortcut (tooltip: *HTTP/REST Cue*)
- **Inspector tabs: Basics**, **Triggers**, **Settings**
- **Targeting:** none — the destination is the URL on the Settings tab.

The Settings tab

[Illustration to be added in a future revision]

Field	Options / range	What it does
URL	any URL	Where the request is sent
Method	`GET`, `POST`, `PUT`, `DELETE`	The HTTP verb
Content Type	editable dropdown: `application/json`, `text/plain`, `application/xml`, `application/x-www-form-urlencoded`	The Content-Type header
Timeout (s)	1–60	How long to wait for a response
Headers	multiline — *Key:Value pairs, one per line:*	Extra request headers
Body	multiline (visible only for POST and PUT)	The request body
Test Request	button + result box	Sends the request now and shows the response

Worked example: switching on a relay-controlled work light

A network power relay at `192.168.1.50` exposes a REST endpoint that takes a JSON POST.

1. Click **HTTP** in the toolbar.
2. **URL**: `http://192.168.1.50/api/outlet/1` — **Method** `POST` — **Content Type** `application/json`.
3. **Body**: `{"state": "on"}`.
4. If the device needs authentication, add it under **Headers**, one per line, e.g. `Authorization:Bearer mytoken`.
5. Click **Test Request** and read the response in the result box. When it answers cleanly, the cue is ready to sit in your preset sequence.

Gotchas

- **Body** only appears for POST and PUT — GET and DELETE requests have no body by design.
- Headers must be one `Key:Value` pair per line, exactly as the field label says.
- The cue fires the request and moves on; a slow endpoint holds the cue for up to **Timeout (s)** seconds. Keep timeouts short for anything in a timing-critical sequence.
- HTTPS URLs work, but self-signed certificates on cheap hardware may be refused — test with **Test Request** well before the house opens.

OSC cues

An OSC cue sends a single OSC message via UDP or TCP: one address pattern, an optional list of typed arguments, done. It's the most direct way to poke a lighting console, sound console, or media server — and the built-in **Test Send** makes it the easiest to verify.

- **Toolbar button: OSC** — no default shortcut (tooltip: *OSC Cue — send a single OSC message via UDP/TCP*)
- **Inspector tabs: Basics**, **Triggers**, **OSC**
- **Targeting:** none — the destination is the Host/Port on the OSC tab.

(For the other direction — controlling *ATTACCA* over OSC, including its full inbound command set and ports — see *Control & Integration*.)

The OSC tab

[Illustration to be added in a future revision]

OSC DESTINATION

Field	Options / range	What it does
Host	editable dropdown (offers `127.0.0.1` and your machine's network addresses)	Destination IP or hostname
Port	1–65535	Destination port
UDP / **TCP**	radio buttons	Transport — UDP is fast, fire-and-forget (recommended for show control); TCP is a reliable connection required by some media servers

MESSAGE

Field	What it does
Address	The OSC address pattern — must start with `/`
ARGUMENTS + **+ Add Argument**	A repeatable list of typed arguments: each row has a Type dropdown (`Int32` , `Float32` , `String`), a Value box, and a ✕ remove button
PREVIEW	A read-only live preview of the assembled message
Test Send + result box	Sends the configured message immediately and reports the outcome

Worked example: firing a preset on a lighting console

An ETC Eos-family console listens for OSC (enable OSC RX in the console's shell settings; the default receive port is UDP 8000). To fire cue 10 in cue list 1 from ATTACCA:

1. Click **OSC** in the toolbar and name the cue "Eos: LX 10 GO".
2. **Host**: the console's IP (say `10.101.90.101`) — **Port**: `8000` — transport **UDP**.
3. **Address**: `/eos/cue/1/10/fire` — no arguments needed.

4. Check the **PREVIEW** box, then click **Test Send**. The console fires cue 1/10.
5. Now the lighting GO rides your cue stack: pair it with the matching sound cue via **Auto-Continue**, and one press of **GO** takes both departments.

The same shape works for any OSC-capable device — only the address dictionary changes. Check the target device's OSC documentation for its addresses and argument types.

Gotchas

- **Test Send** uses the real sender — it genuinely fires the target device. Test with the rig in a safe state.
- The **Address** must start with `/`; the field's tooltip enforces the same rule.
- Argument *types* matter to many receivers: a console expecting `Float32 1.0` may ignore `Int32 1`, and vice versa. Match the device's documentation exactly.
- If UDP messages vanish silently, confirm the device and ATTACCA are on the same subnet and no firewall sits between them; if the device requires TCP (some media servers do), switch the transport radio button.

Common issues

- **Clicking a toolbar button opens a "Pro Feature" dialog instead of creating the cue** — every cue type in this chapter requires ATTACCA Pro. Enter a license key via the dialog's **Enter License Key** button.
- **A Network cue (from a legacy workspace) fires but nothing arrives** — expected in this release: new Network cues can't be created (the toolbar button is disabled), and network output patches have no editor UI yet, so a legacy cue's **Patch** number has nothing to resolve to (a warning is written to the log). Use an OSC, HTTP, or Serial cue instead.
- **MIDI or MIDI File cue does nothing** — no MIDI output device is enabled. Open **Preferences → MIDI**, check **Enable MIDI output (send)** and the device's own checkbox; it becomes the cue's **Destination**/**Patch** automatically.
- **The MIDI cue's Learn button is disabled** — Learn captures from a MIDI *input*, and no input device is enabled in **Preferences → MIDI**.
- **A program change lands one patch off** — the 0–127 wire numbering vs. the 1–128 front-panel numbering. Enter the displayed program minus one.
- **Timecode cue runs but nothing chases it** — frame-rate mismatch (including drop vs. non-drop) between the cue's **Framerate** and the chasing device, or (for LTC) the **Cue Output** channel isn't the one physically connected.
- **Script stops with "Script timed out"** — the run exceeded the **Timeout:** value (default 5 s). Raise it (up to 600 s); remember `wait()` counts.
- **Serial Test Send errors or the device ignores the command** — the COM port is held by another program, or the device wants a terminator you didn't append (**Append CR** / **Append LF**), or parity/stop-bit codes don't match the manual.
- **HTTP cue stalls a sequence** — an unreachable URL holds the cue until **Timeout (s)** expires. Shorten the timeout and verify with **Test Request**.

- **OSC Test Send works from the desk, but nothing during the show** — the console was reachable in rehearsal but is now on a different network/firewall, or the cue is disarmed. The message path is identical for Test Send and GO, so if Test works now, GO will too.
- **A Light cue's channels stay up after the cue ends** — by design: DMX levels hold until another cue changes them. Program an explicit blackout cue.
- **Levels-tab edits are audible/visible immediately (or not at all)** — check the **LIVE EDIT** checkbox state on the Levels tab: on = changes go straight to the wire; off = changes are only heard when the cue fires.

Audio in Depth

Audio cues are the workhorse of most shows, and ATTACCA's audio engine is built around one promise: what's in your file is what comes out of your speakers. This chapter covers everything beyond the basics — choosing an output device, understanding levels, multi-channel files, markers, effects, ducking, playback rate, and fades.

Choosing your output device

ATTACCA plays through a single Windows audio output device, selected app-wide.

1. Open **Edit → Preferences...** (`Ctrl+,`) and select the **Audio** page.
2. The **Audio Output Device** list shows every playback device Windows knows about, each with its name, its type and channel count (for example "WASAPI • 2 channels"), a **Default** badge on the Windows default device, and a checkmark on the device ATTACCA is currently using.
3. Click a device to switch to it. Audio device changes take effect immediately — you do not need to click **Save**, and the change is not undone by **Cancel**.
4. If you've just plugged in an interface and it isn't listed, click **Refresh Devices**.

> *[Illustration to be added in a future revision]*

There is a faster route during rehearsal: the **View → Audio Output Device** submenu lists the same devices and switches immediately (available in Edit Mode).

> *[Illustration to be added in a future revision]*

The same page has an **Audio Latency** section with a **Latency Mode** dropdown: **Low (50ms)**, **Medium (100ms)**, or **High (200ms)** (default Medium). Lower latency makes GO feel snappier; higher latency is more forgiving on busy machines. Unlike the device selection, latency changes take effect on the next app restart.

ASIO is not supported. An earlier build listed ASIO devices, but selecting one didn't route audio correctly, so the option was removed rather than left broken. ATTACCA outputs through standard Windows audio (WASAPI/DirectSound) via the single device you select above.

Levels: what 0.0 means

Every Audio cue has a **Levels** tab with a **Main Level** fader and a matching dB text field. The range is −60 dB to +12 dB, and the text field also accepts `-INF` for full silence. Two buttons give you quick presets: **Set Default Levels** (everything back to 0 dB) and **Set All Silent** (everything to −INF).

> *[Illustration to be added in a future revision]*

A level of **0.0 dB means unity gain** — the file plays back at exactly its recorded level, sample for sample. Negative values attenuate. And although the fader travels up to +12 dB, ATTACCA will not push a signal above the file's own level: anything above 0.0 currently plays at unity. There is no hidden headroom trick — if you

need a quiet file louder, fix it in your editor or DAW.

> **Why does ATTACCA sound louder than Spotify at 0.0?** ATTACCA plays your audio files at their true recorded level. A cue level of 0.0 dB means unity gain: every sample in the file reaches your audio device untouched — no normalization, no compression, no limiting, and no "enhancement" is ever applied unless you explicitly add an effect to a cue. Consumer players like Spotify, Apple Music, and YouTube apply *loudness normalization* by default, typically turning modern masters **down** by 3–8 dB (Spotify targets about −14 LUFS) so all songs sound similar in level. Professional show playback must not do this — your sound designer's levels, and the mastering of your source material, are reproduced exactly. If ATTACCA sounds louder than a streaming app playing "the same" track, the streaming app is quieter than the file, not ATTACCA louder. To match a house-calibrated level, set the cue or output level in ATTACCA — starting from a known unity reference is precisely the point.

Matching your house level

Because 0.0 is a true unity reference, calibrating to your room is straightforward:

1. Play a reference cue at 0.0 and set your amplifier or console trim to the house's normal performance level.
2. From then on, treat 0.0 as "full mix level" and set individual cues down from there — underscoring at −12, preshow music at −18, and so on.
3. If everything in your show feels uniformly hot, turn down the system, not sixty cues. Keep ATTACCA's levels as artistic decisions, and let the rig own the room's overall volume.

Also check the Windows mixer once during setup: ATTACCA never touches its own per-app volume slider, so if someone has pulled it down in Windows **Sound settings** → **Volume mixer**, every cue will be quiet and no fader in ATTACCA will explain why.

Multi-channel audio files

ATTACCA plays multi-channel files (4.0, 5.1, 7.1, and other layouts) **without downmixing** — every channel in the file goes out as-is. Which speaker each channel reaches follows the **Windows speaker configuration** for your output device, not anything set inside ATTACCA. To check or change it, open Windows **Sound settings**, select your output device, and set its speaker configuration (on most systems: Settings → System → Sound → device properties, or the classic Sound control panel → **Configure**). If your interface is configured as stereo in Windows, a 5.1 file cannot reach six outputs.

You can see what ATTACCA found in the file in two places:

- The **waveform** on the **Time & Loops** tab draws **all channels as stacked lanes** — a 5.1 file shows six lanes.
- The **Levels** tab's **Channels:** readout shows the file's true channel count.

[Illustration to be added in a future revision]

[Illustration to be added in a future revision]

What ATTACCA does **not** have yet is per-channel routing. The right side of the Levels tab says exactly this: **"Per-output levels are coming in a future update."** There is no routing matrix in this release — the Main Level fader controls the whole cue, and channels map to speakers only via the Windows configuration described above. A full matrix mixer (per-output levels, crosspoint routing, and per-output Fade cue targets) is planned for a post-launch update. The **Clear Mutes** and **Clear Solos** buttons on this tab are disabled placeholders for the same reason.

Audio markers

Markers are named timestamps inside an audio file. They give you two things: a live countdown in the cue list, and the ability to fire *other* cues at exact musical moments.

Audio markers are included in **ATTACCA Standard** and **Pro**. On the Free tier, markers saved in a workspace are preserved untouched — you simply can't add or use them until you upgrade. See *Licensing & Activation*.

> *[Illustration to be added in a future revision]*

Adding and editing markers

All marker work happens on the Audio cue's **Time & Loops** tab:

- `Ctrl+Click` **on the waveform** adds a marker at that position.
- The **+ Marker** button adds a marker at the current playhead position.
- **Right-click the waveform** for a context menu: **Add Marker Here**, **Rename Marker**, **Delete Marker**, and **Go to Position**.
- **Drag a marker** on the waveform to move it.
- `Ctrl+Wheel` zooms the waveform (1x–20x, same as the **Zoom** slider) for precise placement.

Below the waveform, the **Markers** grid lists every marker with **#**, **Name**, and **Time** columns. Name and Time are directly editable (time in `MM:SS.xx`), and the **X** button deletes a marker. All marker operations are undoable.

(`Shift+Click` on the waveform adds a *slice* marker instead — slices divide the file for playback purposes and are separate from named markers.)

The "Until" countdown column

While an Audio cue with markers is playing, the cue list's **Until** column shows a live `MM:SS.xx` countdown to the next marker. During a vamp or underscore, this is your "time until the next thing happens" clock — many operators run entire transitions off it.

> *[Illustration to be added in a future revision]*

Firing cues from markers

Any cue can be triggered when an Audio cue's playback crosses a marker. On the cue you want to fire, open the **Triggers** tab, check **Audio Marker**, then pick the **Source** (the audio cue that contains the marker) and the **Marker** itself. A preview line summarizes the trigger in plain language.

A worked example — lights up on a musical hit:

1. Your preshow music is Audio cue 1. Add a marker named `Lights Up` at 0:45, right where the downbeat lands.
2. Create a Light cue for your opening look.
3. On the Light cue's **Triggers** tab, check **Audio Marker**, set **Source** to cue 1 and **Marker** to `Lights Up`.
4. GO cue 1. At exactly 0:45 of playback, the lights fire — every night, frame-accurate, no operator timing required.

[Illustration to be added in a future revision]

Marker triggers fire only on **natural playback** crossing the marker. Seeking past a marker (clicking around the waveform in rehearsal) does not fire it, so you can scrub freely without setting off your show. On looping cues, each marker fires once per pass.

Audio effects

Every Audio cue has an **Effects** tab hosting a per-cue effect chain. Eight effects are available, added with the toolbar buttons **+ EQ**, **+ Reverb**, **+ Comp**, **+ Delay**, **+ Chorus**, **+ Echo**, **+ Gate**, and **+ Limiter**:

Effect	Key parameters
Parametric EQ	Interactive curve; per-band Freq (20 Hz–20 kHz), Gain (±15 dB), BW (0.1–10 oct); **+ Band** to add bands
Reverb	Mix, Room, Width, Damping (all 0–100%)
Compressor	Thresh (−60..0 dB), Ratio (1–20:1), Attack, Release, Makeup
Delay	Time (10–2000 ms), Feedback (0–90%), Mix
Chorus	Depth, Rate (0.1–10 Hz), Mix, Feedback, Delay
Echo	Time, Feedback, Mix, Ping-Pong
Gate	Thresh, Attack, Hold, Release, Range
Limiter	Ceiling (−12..0 dB), Release

Each effect in the chain has an enable checkbox, **S**/**L** buttons to save and load a single-effect preset, **^**/**v** to reorder, and **X** to remove; whole chains save and load with **Save Chain** / **Load Chain**. You can edit effect parameters live while the cue is playing.

[Illustration to be added in a future revision]

Effects are **per-cue and opt-in only**. Adding an effect to one cue changes nothing about any other cue.

Automatic protection — only when you use effects

Effects can add gain (EQ boosts, compressor makeup, resonant feedback), so the moment a cue has effects, ATTACCA quietly adds protection:

- With **one or more** enabled effects, a soft-clip stage sits at the end of the chain. It engages only above roughly 0.8 amplitude (about −2 dBFS) and rounds off would-be overs instead of letting them crack; below that level it passes audio through untouched.
- With **two or more** enabled effects and no Limiter of your own in the chain, an automatic safety limiter is added at −0.5 dBFS (very fast, very high ratio) as a final backstop. Add your own **+ Limiter** and the automatic one steps aside in favor of your settings.

> **Why it works this way:** a plain Audio cue is never touched — no limiter, no soft-clip, nothing in the signal path but your file at your level. Protection exists only because *effects* can create levels that never existed in the file, and a digital-over through a PA is a genuinely dangerous noise. The moment you remove the effects, the protection is gone too.

Ducking other cues

On any cue's **Triggers** tab, the Duck/Boost section lets a cue push down everything else when it fires — the classic "announcement over preshow music" move:

- **Duck/Boost audio of other cues** — the enable checkbox.
- **Level** — the level, in dB, that every *other* running Audio cue is set to when this cue starts (negative to duck, e.g. −20).
- **Time** — a ramp-time field. In this release the change is applied **instantly**; the Time value is not yet honored.

[Illustration to be added in a future revision]

Be aware of how this actually behaves: the duck is applied once, when the ducking cue starts, and **it is not automatically restored** when that cue ends. The other cues stay at the ducked level until something changes them.

The reliable pattern is to restore levels yourself:

1. Cue 10 — your announcement, with **Duck/Boost audio of other cues** checked and **Level** −20.
2. Cue 11 — a **Fade** cue targeting the music cue, fading its Main Level back to its performance value over 2–3 seconds, set to auto-follow cue 10.

This gives you a clean duck *and* a musical recovery, on your own timing. (If you want the music gone rather than ducked, the **Fade & Stop Others** behavior on the same tab fades and stops other cues in a chosen scope instead.)

Playback rate and pitch

The **Time & Loops** tab's **Rate:** field sets playback speed anywhere from **0.03 to 33.0** (1.0 = normal), and it applies live to a running cue.

By default, changing the rate changes pitch too, exactly like varispeed on a tape deck — 2.0 plays twice as fast and an octave up. Check **Preserve pitch** and ATTACCA switches to time-stretching: the tempo changes but the pitch stays put, at the cost of slightly more CPU.

A quality note: pitch-preserved stretching is excellent for modest changes (roughly 0.7–1.4), where it's ideal for nudging a music bed to fit a scene change. At extreme rates the time-stretching algorithm audibly smears transients — for drastic speed effects, plain varispeed (Preserve pitch off) usually sounds better, pitch shift and all.

Fades

Audio cues have a built-in fade envelope, separate from Fade cues:

1. On the **Time & Loops** tab, check **Integrated fade**.
2. Set **In:** and **Out:** times in seconds.
3. Optionally check **Lock fade to cue** to pin the envelope to the cue's timeline.

The fade-in ramps from silence to the cue's level over the In time, starting at the start time. The fade-out **leads into the end of the cue**: it begins Out-seconds before the cue's end time and completes *at* the end time (the file's end, or the **End time:** you set). You never hear the audio stop abruptly after a completed fade, and you never lose program material to a fade that starts at the end — set a 5-second Out and the last 5 seconds of the cue are the fade.

Integrated fades are always linear. For shaped fades — S-curve or parametric, in your choice of level domain — and for fading a cue that's already running, use a **Fade** cue targeting the Audio cue (see the Fade cue coverage in *Working with Cues*). Fade cues are also how you fade rate, and how you restore levels after a duck.

[Illustration to be added in a future revision]

Common issues

- **Everything is louder than my reference tracks from a streaming app** — that's the streaming app's loudness normalization turning masters down, not ATTACCA adding gain. See "Levels" above; ATTACCA at 0.0 plays the file exactly.
- **A 5.1 file only comes out of two speakers** — the Windows speaker configuration for your output device is set to stereo. Fix it in Windows Sound settings, not in ATTACCA.
- **Cue is quiet no matter what the fader says** — check the Windows Volume mixer; ATTACCA's per-app slider may have been pulled down outside the app.
- **Raising Main Level above 0.0 doesn't get louder** — by design in this release, levels above unity clamp at the file's own level. Re-master the file louder if you need more.
- **Music never comes back after an announcement** — ducking doesn't auto-restore. Add a Fade cue after the ducking cue to bring levels back (see "Ducking other cues").
- **Markers fire in rehearsal when I click around the waveform** — they shouldn't and they don't: seeking past a marker never fires it. If a cue fired, playback genuinely crossed the marker.

- **Effects sound different at high level** — with effects present, the protective soft-clip engages above ≈0.8 amplitude. Pull the effect gains or the cue level down a little, or add your own Limiter with a lower ceiling.
- `-INF` **won't type in the dB field** — it does; type the minus sign then `INF` (the field accepts `-INF` for silence).
- **New audio interface isn't listed** — click **Refresh Devices** on the Audio preferences page; if Windows doesn't see the device, ATTACCA can't either.
- **Rate/pitch change sounds grainy** — large pitch-preserved rate changes smear audibly. Try Preserve pitch off, or a smaller rate change.

Video & Camera in Depth

Video and Camera cues (included in Standard and Pro) put moving images on any display connected to your machine — projections, scenic loops, IMAG, confidence feeds. This chapter covers display mapping, geometry, color, playback behavior, supported formats, the output window itself, and live cameras.

Displays and assignment

Display Mapping: logical displays, physical monitors

ATTACCA separates *which display a cue targets* from *which physical monitor that is*. Cues reference logical display numbers (Display 1, Display 2, …); a workspace-level table maps those numbers to real monitors. When you move to a different venue, you re-point the mapping once — no cue editing.

1. Open **Edit → Preferences…** (`Ctrl+,`) and select the **Displays** page (header: **Display Mapping**).
2. Each row is one logical display: its number, an editable **Name** (this name appears in the cue inspector's display dropdown — call it "Projector" or "US Wall", not "Display 2"), the **Physical Monitor** it maps to, and a **Fullscreen** checkbox.
3. **+ Add Display** and **- Remove Display** manage rows (minimum one).
4. Uncheck **Fullscreen** to get a movable, resizable output window instead of a full-monitor one — essential on single-monitor setups, where a fullscreen output would cover the cue list.

[Illustration to be added in a future revision]

The mapping is saved in the workspace file, so a show file carries its display plan with it — including two per-display options covered in full in *Settings Reference*: **Standby Black**, which keeps a black backer on a display so the desktop never appears between cues, and a **custom output region**, which confines the output to a sub-rectangle of the monitor for LED processors and hard-edged surfaces. Each physical monitor may be the target of only one mapping.

Output resolution, and why there is no setting for it

ATTACCA does not choose an output resolution — it outputs at whatever resolution Windows has negotiated with the display or processor. If an LED processor or projector needs a specific mode, set it in **Windows Display Settings** first, then map to it here. There is also **no hot-plug detection**: after connecting or reconnecting a monitor, reload the workspace so the mapping resolves against the current monitor set.

Assigning a cue to a display

On a Video cue's **I/O** tab (and a Camera cue's **Camera** tab), the **Video Output** section has a **Display:** dropdown listing your mapped displays as "Display N — Name", plus a **Display #:** field for the stage number. Pick the display; the cue's output window opens on the mapped monitor at GO.

[Illustration to be added in a future revision]

When a display is missing

Show rigs change. If a cue targets a display whose monitor isn't currently connected:

- **Your assignment is never lost.** The cue keeps its saved display assignment in the workspace file, untouched.
- **At load and at GO, ATTACCA falls back for the session.** It resolves the monitor by saved **name** first, then by Windows **device name**, then by **position** (the Nth monitor), and finally to the **first available display** — and writes a warning to the log saying what it did. If the workspace's default video display is missing at load, the app likewise uses the primary display for that session only.
- **The fix is a re-fire.** Plug the display back in (or arrive at the venue with the right monitor connected) and fire the cue again — each GO re-resolves against the current monitor set. Nothing needs re-saving.

[Illustration to be added in a future revision]

Mixed-DPI rigs

If your monitors use different Windows scaling factors (say 150% on a laptop, 100% on a projector), the video output on the second monitor may look **slightly soft** — the window is rendered at the main window's DPI and stretched by Windows. Geometry stays correct: positions, sizes, and crops land where you put them. Per-monitor sharpness is planned for a post-launch update. For maximum crispness today, run your show displays at the same scaling factor.

Geometry

A Video cue's **Geometry** tab controls where and how the image sits on the stage. (Camera and Image cues have the identical tab.)

- **Mode:** — **FillStage** (default) sizes the image to the whole display, or **Custom**, which unlocks the full Transform section.
- **Fill Style:** — with FillStage: **Fit** (letterbox, whole image visible), **Fill** (cover the stage, overflow cropped), or **Stretch** (distort to fit exactly). A fourth entry, **Native**, appears in the list but is **not applied by this release's renderer** — a cue set to Native renders as **Fit**. The value is saved with the cue. (Note that a workspace saved with Native will not open in ATTACCA builds older than this one.)
- **Opacity:** (0–1), **Layer:** (0–1000, higher on top), and **Blend Mode:** (23 modes from Normal through Additive, Multiply, Screen, Overlay, and friends).
- In **Custom** mode: **Position X/Y** (±5000 px), **Scale X/Y** (0.01–5.0, with a 🔒 button to lock them together), **Rotation** on all three axes (±360°), **Anchor X/Y** (0–1, with **Center** / **TL** / **TR** / **BL** / **BR** preset buttons), and **Crop** Top/Bottom/Left/Right in pixels.
- **Preserve Aspect Ratio** and **Smooth (Bilinear Filtering)** checkboxes round out the tab.

Every numeric row is a slider plus a numeric box plus a ↺ reset button — and you can type values beyond the slider's range when you need to.

[Illustration to be added in a future revision]

Two controls on this tab are disabled in this release. Blend Mode and **Rotation X/Y** are not applied by the current renderer, and their controls say so on hover: *"Coming in a future update — … is not yet applied by the current renderer. The value is saved with the cue."* Rotation **Z** works normally, and any values previously set for the disabled controls are preserved in the file.

[Illustration to be added in a future revision]

Geometry is stored on the cue and **applied every time the cue plays**. Stop a cue, re-fire it, fire it from a different point in the show — it comes back exactly where you built it. (Earlier builds could lose geometry on a retrigger; that's fixed.)

Color adjustments

The Video cue's **Effects** tab holds live color controls — all of them apply in real time while the cue plays:

Control	Range	Normal
Brightness:	−1 to +1	0
Contrast:	0 to 3	1
Saturation:	0 to 3	1 (0 = grayscale)
Blur:	0 to 50	0 (off)

Set Saturation to 0 for an instant black-and-white treatment. There is **no gamma control** — if your blacks look grey on a projector, the usual culprit is the GPU's output dynamic range setting (set it to Full, not Limited, in your GPU control panel), not a missing gamma knob.

[Illustration to be added in a future revision]

Playback: loops, endings, and fades

The **Time & Loops** tab controls the clip's life cycle:

- **Start Time:** / **End Time:** trim the playable region.
- **Rate:** — playback speed, 0.03–33.0.
- **Play Count:** — how many times the clip plays (minimum 1). A count of 3 plays exactly 3 passes.
- **Infinite Loop** — loops until stopped.
- **Hold Last Frame** — see below.

Fades live on the **I/O** tab's **Transition** section: **Fade In:** ramps opacity from 0 to the cue's target on GO; **Fade Out:** ramps it back to 0 (0 = instant in both cases).

Fade-out lead-in. A Video cue's built-in fade-out *leads into* the end of the clip: it begins Fade-Out-seconds before the effective end and **completes at the cue's end time** — not after the last frame has already frozen. Set End Time 1:00 with a 3-second fade-out, and the picture fades over 0:57–1:00, reaching black exactly as the cue ends. On looping cues, the lead-in arms only on the final pass.

What happens at the natural end vs. Stop:

- **Natural end, Hold Last Frame unchecked:** the fade-out (if any) leads into the end, then the output closes.
- **Natural end, Hold Last Frame checked:** the clip **freezes on its final frame** and stays on screen — until the cue is stopped, another visual cue takes over the display, or panic. The fade-out lead-in deliberately does not arm; the hold would be pointless if the image had already faded to nothing.
- **Stopped early:** the Fade Out time is applied from the moment of the stop — opacity ramps to 0, then the output closes. Panic uses the panic fade time instead (see *Getting Started* for panic behavior).

[Illustration to be added in a future revision]

[Illustration to be added in a future revision]

> **Why it works this way:** "hold last frame" is the standard trick for ending a video on a scenic image — the clip becomes its own still. If the fade-out ran first, you'd hold three seconds of black.

Supported formats

ATTACCA ships its own video decoders — nothing to install, and installing codec packs changes nothing about what plays.

Containers (the file picker offers `*.mp4 *.mov *.avi *.mkv *.wmv *.webm *.m4v`):

Container	Notes
MP4 / M4V / MOV	The primary show-content formats
MKV / WebM	Fully supported
AVI	Legacy content; also the usual HAP/MagicYUV carrier
WMV / ASF	Supported
MPEG-TS / MPEG-PS	Broadcast-origin content

Video codecs: H.264/AVC, H.265/HEVC, ProRes (all 422 profiles and 4444, including alpha), the HAP family (HAP, HAP-Q, HAP-Alpha, HAP-R), MagicYUV, MPEG-1/2/4, VC-1/WMV, VP8/VP9, AV1, DV, DNxHD/DNxHR, CineForm, QuickTime Animation, PNG- and MJPEG-in-MOV (alpha loops), and Theora.

Audio inside video files: AAC, MP3/MP2, PCM, AC-3/E-AC-3, Vorbis, Opus, FLAC, ALAC, and WMA all play, with per-cue volume and mute on the Video cue's **Audio** tab.

Still images (`.png .jpg .jpeg .tiff .tif .bmp .gif`) can be given to a Video cue and display as stills — though the dedicated **Image** cue, with its hold duration and fades, is usually the better tool.

Honest notes for content prep:

- **Decoding is software-only** — the GPU is not used for decode. H.264 and ProRes are light; 4K H.265 is CPU-hungry, and several simultaneous 4K HEVC clips will tax any machine. For heavy multi-layer content,

HAP or ProRes trade disk space for very low CPU cost — that's what they're for.

- **H.264 in MP4** is the safe default for general content.
- **Local files only.** Video cues play validated local file paths — network streams and URLs are not supported. Copy media onto the show machine (and keep it with the workspace so relative-path relinking works — see the missing-media coverage in *Working with Cues*).
- Exotic or damaged files that other players "sort of" play may not; when in doubt, transcode to H.264/MP4 or ProRes/MOV.

The video output window

When a video, image, text, or camera cue fires, ATTACCA opens an **output window** on the target display — borderless, fullscreen (when the mapping says so), and kept on top of other windows. One output window serves each display; cues targeting the same display share it (by default a new visual cue replaces the previous one — see the overlap setting in Preferences → Video).

[Illustration to be added in a future revision]

Operator notes worth memorizing:

- **The output window is not a control surface.** It accepts no typing and no ordinary shortcuts, with one deliberate exception: `Esc` pressed while an output window has focus triggers a full panic (and the window fades out and closes).
- **Keep your focus on the main window.** If you click into an output window, most of your keyboard — GO, stop, everything except `Esc` — is disconnected from ATTACCA until you click the main window again.
- `Ctrl+Shift+F12` **is the emergency exit.** It's a system-wide hotkey that closes every video output window, and it works even if the app's UI is unresponsive.
- On a **single-monitor system**, the output window is deliberately *not* kept on top (unless the mapping forces fullscreen), so a windowed output can't trap you away from the cue list.

Camera cues

A Camera cue puts a live camera feed on a display, with the same geometry controls as video.

Setting up

On the cue's **Camera** tab:

- **Device:** — a dropdown of detected cameras (webcams, USB capture devices); you can also type a device name. The refresh button re-scans.
- **Resolution:** and **FPS:** — request a capture mode (1920x1080 / 1280x720 / 3840x2160 / 640x480 presets, or type your own; 15–60 fps). The camera delivers the nearest mode it supports.
- **Status:** — a colored dot: **Ready** (green — warmed and waiting), **Live** (on an output now), **Warming…**, **Not connected**, or **Idle**.
- **Display:** / **Display #:** and **Fade In:** / **Fade Out:** work exactly as on Video cues.

The **Geometry** tab is identical to the Video cue's — scale, position, crop, and blend a camera like any other source.

[Illustration to be added in a future revision]

Warmup and the instant GO

Cameras are slow to open — physically, the device takes 2–3 seconds to start streaming. ATTACCA hides that cost entirely: **every camera referenced by a Camera cue is opened when the workspace loads** and kept running, warm and invisible. GO simply redirects the already-running feed to the output window, so the first frame appears essentially instantly.

Two things follow from this, and both are normal:

- **Your camera's LED is on from workspace load until the workspace closes** — not just while a camera cue is running — and the device consumes USB bandwidth the whole time. That's the price of an instant GO, and there is currently no preference to opt out. If a lit camera light matters in your space (or you simply don't want a live device), unplug the camera or remove the Camera cues from the workspace.
- The log line `FlashCapCameraCapture: frames arriving — render suppressed (invisible warmup target)` just means a warmed camera is alive and intentionally not being rendered. It is not an error.

Panic closes the camera's output window, but the warm feed survives — you can re-fire instantly.

One output per camera

A single camera device drives **one output at a time**. You cannot send the same camera to two displays simultaneously — firing a second cue that uses the same device takes the feed rather than duplicating it. To cover two displays with camera at once, use two physical devices.

Do not hot-unplug displays mid-show

ATTACCA does not detect a monitor disconnecting **during** a session. If a show display is unplugged (or sleeps, or renegotiates) while an output window is live on it, Windows decides what happens to the window — usually it relocates it, which can drop a fullscreen, always-on-top video window right on top of your cue list.

If it happens:

1. **Click into the stray video window** to give it focus.
2. Press `Esc` — full panic; the window fades and closes. (`Ctrl+Shift+F12` also closes all output windows from anywhere.)
3. Reconnect the display, then **re-fire the cue** — every GO re-resolves displays against the current monitor set, so the new window lands correctly.

If Windows *doesn't* relocate the window, the video simply keeps playing offscreen — audio continues and the cue completes invisibly; a re-fire brings it back to a real display.

The real guidance is the heading: lock your display cabling down before the house opens, and don't hot-unplug displays mid-show.

Common issues

- **Video opened on the wrong monitor** — check Preferences → Displays (Display Mapping): the logical display the cue targets is probably mapped to a different physical monitor at this venue. Re-map once; re-fire.
- **Output is slightly blurry on the second monitor** — mixed Windows scaling factors; see "Mixed-DPI rigs". Match the scaling on both displays for full sharpness.
- **Blacks look grey on the projector** — GPU output range, not ATTACCA: set your GPU's output dynamic range to Full (e.g. NVIDIA Control Panel → Change Resolution → Output dynamic range → Full). Common on HDMI.
- **4K clip stutters** — decoding is CPU-only. Transcode to HAP or ProRes for cheap decode, or to H.264 at a saner resolution.
- **Clip ends on black instead of holding the image** — the fade-out ran; with **Hold Last Frame** checked the fade-out intentionally doesn't arm, so if you see black, the checkbox isn't set on this cue.
- **Keyboard "stopped working" during a video** — you clicked into the output window. Click the main ATTACCA window; only `Esc` (panic) works from inside an output window.
- **A video window is stuck covering my screen** — click into it and press `Esc`, or press `Ctrl+Shift+F12` to close all output windows; then re-fire once your displays are sorted.
- **Camera light is on but no camera cue is running** — normal: cameras warm up at workspace load and stay warm for instant GO. Unplug the device or remove Camera cues if that's not acceptable in your space.
- **Camera takes seconds to appear on GO** — the warm pool couldn't open the device earlier (check the Status dot on the Camera tab; **Not connected** means the device wasn't available at load). Reconnect it and check the status turns **Ready**.
- **Same camera won't show on two displays** — one output per device, by design. Use a second camera.
- **"render suppressed (invisible warmup target)" in the log** — normal warmup behavior, not a fault.

Lighting

ATTACCA is a full DMX lighting controller as well as a sound and video playback system. You can patch fixtures, program looks as Light cues, run chases and color effects on top of them, record DMX from an external console, and monitor every channel going out the wire — all from the same cue list that runs your audio and video.

Lighting is a **Pro** module. On the Free tier, Light and Effect cues appear locked in the cue list and the lighting windows and the **Lighting** preferences page show the upgrade prompt. All lighting menu items live in the **View** menu and are available in Edit Mode.

Lighting concepts for newcomers

If you already speak DMX, skip ahead to *Enabling lighting output*.

Universes and channels

DMX is the language of stage lighting. One DMX **universe** is a stream of 512 **channels**, each holding a value from 0 to 255. A simple dimmer takes one channel (0 = off, 255 = full). A color-mixing LED fixture takes several — for example one channel each for red, green, blue, and intensity. A moving light might take 20 or more.

When one universe's 512 channels aren't enough, you add more universes. ATTACCA can output up to 64 universes over the network at once.

sACN vs Art-Net

ATTACCA sends DMX over your ordinary network using either of two standard protocols. Both carry the same channel data; the difference is how the packets travel:

	sACN (E1.31)	Art-Net
Delivery	**Multicast** — packets are addressed to a group; only devices that subscribe to a universe receive it	**Broadcast** (default) — packets go to every device on the network; or **unicast** to one specific IP
Best for	Larger rigs, shared networks, modern nodes and consoles	Older nodes, simple point-to-point setups, gear that only speaks Art-Net
Extras	Per-source **priority** (0–200) so two senders can share a rig politely	Net/Subnet/Universe addressing scheme

Plain-terms guidance: if your gateway or fixtures support sACN, prefer it — multicast means only the devices that care about a universe see its traffic, which scales better and is kinder to everything else on the switch. Use Art-Net when that's what your hardware speaks, and switch Art-Net to unicast (a single target IP) if broadcast traffic bothers other devices on the network.

You can enable both protocols at the same time; they carry the same universes.

Fixtures and personalities

A **fixture definition** (sometimes called a personality or profile) tells ATTACCA what each channel of a light does — "channel 1 is the dimmer, channels 2–4 are red/green/blue." Many fixtures also offer several **modes** with different channel counts (an 8-channel mode with extras versus a basic 4-channel mode — match the mode set on the physical fixture). **Patching** is the act of telling ATTACCA which fixture sits at which DMX address in which universe. Once patched, you can program by fixture name and color instead of raw channel numbers, and the channel grid labels each channel with its function.

> **Why it works this way:** Network DMX exists because a lighting rig can need dozens of universes, and running a dedicated 5-pin cable per universe from the operator position doesn't scale. Putting DMX on Ethernet lets one network cable carry every universe to a gateway ("node") near the rig, which converts it back to physical DMX for the fixtures. The trade-off is that your lighting now shares rules with every other network device — which is why picking the right network interface matters (below).

Enabling lighting output

Open **Edit → Preferences…** (`Ctrl+,`) and choose **Lighting**, or use **View → Lighting Settings…** to jump straight there. Click **Save** to apply changes.

Output settings belong to this computer, not to the show file. Which protocols are enabled (sACN / Art-Net), broadcast or unicast, multicast or unicast, the network interface, the Art-Net addressing, and the USB-DMX ports are saved per user on this machine (`%APPDATA%\ATTACCA\lighting-output.json`). Opening a workspace never changes them, and saving one never carries them along. A fresh install transmits **nothing** until you enable a protocol here and click **Save** once; after that the setting persists across every show you open. The settings that *do* travel with the show file are **Merge Mode**, **Universe Count** and **Start Universe** (they describe the show), plus the light patch.

> **Why it works this way:** a show file describes the show; it should never describe the network of whichever computer opens it. Before this rule, opening a workspace that had been saved with Art-Net on switched a sACN-only machine to Art-Net broadcast on every network adapter — a file changed where the DMX went. Now there is exactly one place your output is configured, so the rig you tested with is the rig you get, whatever file you open. Workspaces saved by older versions still contain the old output settings; ATTACCA reads past them (it notes in the log that it did) and leaves your Preferences alone.

[Illustration to be added in a future revision]

Before your first light cue

Do this once per computer, before you expect anything to reach the rig — including after installing an update that introduced this rule:

1. Open **Edit → Preferences… → Lighting**.
2. Under **Network Interface**, pick the adapter that is actually wired to your lighting network (click **Refresh** if it isn't listed). Not "all interfaces" on a machine with more than one network.

3. Enable the protocol your rig speaks — **Enable sACN output** *or* **Enable Art-Net output** — and choose its mode (**Multicast** / **Broadcast** for a normal rig; unicast plus a target IP for a single node).
4. Click **Save**.
5. Open any Light cue's Recording tab and read the **Playback output:** line: it should name the protocol and the interface (`sACN multicast — Ethernet 10.101.10.5 — set in Preferences → Lighting`). If it says "none enabled", go back to step 3.

> **Why it works this way — fail-closed by design:** a fresh install transmits nothing until you choose a network. That is deliberate. DMX output is a broadcast onto whatever network the computer is connected to; sending it on a guess (the venue Wi-Fi, the office LAN) is at best noise and at worst somebody else's rig doing what your cues say. So nothing goes out until a person has looked at the list of adapters and picked one. The price is one visit to Preferences per computer; the benefit is that the rig you tested with is the rig you get, whatever show file you open.

Which settings live where. Two kinds of setting are involved and they are stored in different places:

Setting	Lives in	Why
Enable sACN / Enable Art-Net, multicast / broadcast / unicast modes and targets, sACN source name, priority and loopback, the **Interface**, Art-Net Net / Subnet / Universe, USB-DMX devices	**This computer** (`%APPDATA%\ATTACCA\lighting-output.json`), set in Preferences → Lighting	They describe *this machine's* network and hardware. A show file must never change where another computer sends DMX.
Universe Count, **Start Universe**, **Merge Mode**, the **Light Patch**	**The show file** (`.atca`)	They describe the show, and must travel with it.
Per-cue playback settings: trim, loop mode, crossfade, **Fade in** / **Fade out**, **On finish** (and its Go-to cue), the recording input universes	**The cue**, in the show file	They are part of the cue, like a level or a pre-wait.

Opening a workspace never changes the first row; saving one never carries it along. Workspaces saved by older versions still contain output settings — ATTACCA reads past them and notes in the log that it did.

Network Interface

The **Interface** dropdown (with a **Refresh** button) selects which network adapter carries your lighting traffic, for both Art-Net and sACN. In plain terms: if your machine has more than one network connection — venue Wi-Fi, an office LAN, and a dedicated lighting network — this setting picks the network card that is actually wired to your rig. Choose the adapter connected to your nodes and gateways.

If you leave it unselected, ATTACCA binds to all interfaces and lets Windows decide where packets go. That usually works on a single-network machine, but on a multi-network machine it can spray DMX onto the wrong network or send multicast out the wrong port — ATTACCA writes a warning to its log file when it starts up

bound to all interfaces for exactly this reason. On any machine with more than one active adapter, pick the specific interface.

Art-Net

- **Enable Art-Net output** — off until you enable it (a fresh install transmits nothing).
- **Broadcast mode** — on by default. Uncheck it to unicast instead, and enter the node's IP in the **Target Address** field that appears.
- **Art-Net Net** (0–127), **Art-Net Subnet** (0–15), **Art-Net Universe** (0–15) — Art-Net's own addressing scheme, 0-indexed per the Art-Net protocol. Leave these at 0 unless your node's documentation says otherwise.

sACN (E1.31)

- **Enable sACN output** — off until you enable it; turn it on to use sACN. Enabling *both* protocols is legal but unusual — most rigs speak one, and the Recording tab's **Playback output** line says "(both enabled)" as a gentle reminder.
- **Source Name** — the name ATTACCA advertises to sACN devices (default `ATTACCA`).
- **Priority (0-200)** — default 100. If two sACN sources send the same universe, receivers obey the higher priority.
- **Multicast mode** — on by default (the normal way to run sACN). Uncheck to unicast to the IP in **Unicast Target**.
- **Multicast loopback (same-machine monitoring)** — on by default so monitoring tools such as sACN View running on the same computer can see ATTACCA's output. The tooltip recommends disabling it for production.

DMX Universes

- **Universe Count** — how many universes ATTACCA outputs (1–64).
- **Start Universe** — the first universe number (default 1).

DMX Merge

Merge Mode decides what happens when two things (a cue and the dashboard, or two cues) write the same channel: **Htp** (Highest Takes Precedence) or **Ltp** (Latest Takes Precedence — the default, and how most theatrical consoles behave).

USB-DMX (Enttec DMX USB Pro)

If you have an Enttec DMX USB Pro (or compatible FTDI widget), add it here with **Add Device** / **Refresh Ports**: each device card has a Name, COM Port, Universe, an Enabled checkbox, a status badge, and a Remove button. USB devices output their universe alongside the network protocols.

There is no output frame-rate setting: ATTACCA streams DMX continuously at the standard rate (~44 frames per second on the network, ~40 on USB) so fixtures never time out.

Patching fixtures

Patching is a short workflow: browse the Fixture Library to find your fixture, then use the Light Patch window to place instruments at DMX addresses. For a rack of simple dimmers, the Quick Patch dialog does it in one step.

Browsing the Fixture Library

Open **View → Fixture Library...**. The library opens as a picker dialog with two tabs:

- **Built-in Fixtures** — a **Search fixtures...** box, a **Manufacturers** tree (with an **All Manufacturers** entry), a **Fixtures** list, and a detail pane showing the selected fixture's **Mode** dropdown, its **DMX Channels** count, and a **Channels** table (**Ch#** / **Name** / **Type** / **Width** / **Caps**) so you can confirm the profile matches your fixture's manual.
- **Download from OFL** — searches the Open Fixture Library online (search box hint: `Search Open Fixture Library (e.g. "Martin MAC")...`, then **Search** and **Import Selected**). Requires an internet connection.

Pick a fixture and click **Select** (or double-click it), or **Cancel** to close. You will usually reach the library through the **Browse...** button inside the Patch Fixtures dialog rather than opening it on its own.

[Illustration to be added in a future revision]

The Light Patch window

Open **View → Light Patch...** (or click **Edit Light Patch...** at the bottom of Preferences → Lighting — both edit the same patch). The window is a toolbar over a patch grid, with a detail panel and a Groups panel.

Toolbar buttons, left to right: **Quick Patch...** · **Patch Fixtures...** · **+ New Instrument** · **+ New Group** · **+ Group from Selected** · **Auto-patch All** · **Delete Selected**.

The patch grid columns: a status icon, then **Name**, **Address**, **Universe**, **Output**, **Definition**, **Note**. Select a row to edit it in the detail panel: **Name**, **Output**, **Note**, **Universe** (1-based), **Definition**, and **DMX Address** (starting channel, 1–512).

[Illustration to be added in a future revision]

The Patch Fixtures dialog

Click **Patch Fixtures...** to patch one or many of the same fixture in a single pass. Every field:

- **Fixture** — the chosen definition, with a **Browse...** button that opens the Fixture Library.
- **Mode** — the fixture's channel mode, with its channel count shown beside it.
- **Quantity** — how many to patch.
- **Universe** — which universe they go in.
- **Start Addr** — the first fixture's DMX address; the rest follow sequentially.
- **Group Name** — a group is created automatically containing the new fixtures.
- **Naming** — **Numeric (1, 2, 3...)** or **Named (Fixture 1, Fixture 2...)**.
- **Output** — which output the instruments use.

A preview box shows the resulting addresses (or an error if they don't fit). The footer reminds you: "Fixtures will be patched with sequential DMX addresses. A group will be created automatically." Click **Apply** to patch or **Cancel** to back out.

[Illustration to be added in a future revision]

The Quick Patch dialog

Quick Patch… is the fast path for generic instruments (dimmers, simple RGB pars): **Fixture Type**, **Mode**, **Quantity**, **Universe**, **Start Addr**, **Naming** (**Numeric (1, 2, 3…)** or **Named (Dimmer 1, Dimmer 2…)**), and **Output**, with a preview box. The footer note explains the payoff: "Instruments will be patched with sequential DMX addresses. Use numeric naming for commands like '1 = 255'." Click **Patch** to apply.

[Illustration to be added in a future revision]

Editing and removing patches

- Rename an instrument inline via right-click → **Rename**, or edit any field in the detail panel.
- Remove instruments with the **Delete Selected** toolbar button or right-click → **Delete**.

Deletes are immediate — there is no confirmation dialog. Deleting an instrument or a group takes effect the moment you click. Double-check your selection before deleting, especially mid-tech.

Fixture groups

Groups let you address several fixtures at once ("backlight = 50") and drive fixture selection in the color picker and Effect cues.

- The **Groups** panel lists all groups, with **Delete** and **Rename…** buttons (Rename opens a small dialog: "New name:") and a **Members** checklist for the selected group.
- Create groups with **+ New Group**, **+ Group from Selected** (tooltip: "Create a new group containing the currently-selected instruments."), or automatically via the Patch Fixtures dialog's **Group Name** field.
- Right-click rows in the patch grid for **New Group from Selection**, and the **Add to Group** and **Remove from Group** submenus, which list every existing group — the quickest way to reorganize membership.

Light cues

A Light cue sets levels. Create one with the **Light** toolbar button (`Ctrl+6`). Its inspector tabs are **Basics**, **Triggers**, **Levels**, **Curve**, and **Recording**.

A Light cue can be programmed two equivalent ways, and they work together:

- **Levels** — set channel values directly in the grid, on faders, or with the color picker.
- **Commands** — type lines like `1 = 100` or `backlight.zoom = 50` in the **Light commands:** box at the bottom of the Levels tab (one command per line; invalid lines show a red validation error). Commands can address channels, fixtures, groups, and fixture parameters by name.

When the cue fires (**GO**), its levels are written to the DMX output and fade in over the time set on the Curve tab.

The Levels tab

[Illustration to be added in a future revision]

Top to bottom:

- **DMX Channel Grid** with a **Universe:** dropdown — a 16-column grid of every channel in the selected universe. Each cell shows the channel number, an editable value (0–255), and a parameter label (Dimmer, Red, Pan…) read from your patch.
- **Select Range** — **Start:** / **End:** / **Value:** boxes and a **Set All** button to bulk-set a channel range.
- **Channel Faders (used channels)** — a vertical fader for every channel the cue uses; drag to set.
- **Preset Colors (for LED fixtures)** — one-click **Red**, **Green**, **Blue**, **White**, **Amber**, **Warm**, **Cool** buttons.
- **Color Picker** — the heart of fixture-aware programming (next section).
- **Blackout All** — sets every channel to 0 across all universes in this cue.
- Display options: **View mode** (**Slider** / **Tile** / **Text**), **Show as %** (percentages instead of raw 0–255).
- **Collate effects of previous light cues** — combine this cue with looks from earlier light cues rather than replacing them.
- **Use as subcontroller in dashboard** — exposes this cue as a fader in the Light Dashboard's SUBCONTROLLERS strip.
- **Light commands:** — the multiline command box described above.

The channel grid's parameter labels are read from the patch when the tab opens or when you switch universes. If you edit the patch while a Levels tab is open, the new labels appear the next time you switch universes or reselect the cue — not instantly.

The color picker and fixture selection

When your patch has fixtures, the Color Picker section shows a fixture chooser: a **Group:** dropdown, **All** / **None** buttons, and a checkbox row per fixture (name, address, capability) with a selection summary. Below it:

- A full color wheel with **R** / **G** / **B** sliders and a **#** hex field.
- **Quick swatches** — 14 one-click colors: Warm White, Cool White, Red, Green, Blue, Amber, Cyan, Magenta, Purple, Orange, Deep Blue, Lavender, Congo Blue, and No Color / Blackout.
- **My presets** — your own saved colors. Click **Save current…** and name it; right-click a saved preset for **Delete**.
- **Gels (Rosco / Lee)** — an expander with the gel library: filter by manufacturer, search by name or number, and click **Apply** on a gel to use its color.
- **Intensity:** slider (0–100%) for the selected fixtures, and **Pan:** / **Tilt:** sliders (0–255, centre 128, with reset buttons) when the selected fixtures have movement.

With no fixtures selected, the picker falls back to manual mode: choose a **Layout:** (**RGB**, **RGBW**, **RGBA**, **CMY**), a **Start ch:**, and (for layouts with white) a **White mix:** slider.

[Illustration to be added in a future revision]

Live Edit

Normally, edits to a Light cue take effect only when the cue fires. Tick the **LIVE EDIT** checkbox at the top of the Color Picker section to push values to the rig as you program: every fader move, swatch click, and gel pick goes straight to DMX output. While it's on, a red ● and the words **(live to wire)** appear beside the checkbox.

Live Edit is the natural way to focus and build looks with the actual rig in front of you — but remember that the audience sees everything you touch. Turn it off before making edits during a performance.

Fade times (the Curve tab)

The **Curve** tab controls how the cue's levels fade in:

- **Curve type:** — **Custom**, **Linear**, **SCurve**, or **Parametric**.
- **Duration (seconds):** — the fade time.
- **Parametric intensity:** — curve shaping (visible in Parametric mode).
- **Use split fade times (separate up/down)** — reveals **Up fade:** and **Down fade:** boxes so channels moving up can fade at a different speed than channels moving down, console-style.

Recorded cues: the Curve tab is greyed out. When a Light cue has a recording attached, the controls above are disabled and the tab shows one line: *Fade curves apply to fixture-based Light cues. Recorded playback uses Fade In / Fade Out on the Recording tab.* Detach the recording (✕ **Clear** on the Recording tab) and the controls come back with their old values.

> **Why it works this way:** a curve shapes how ATTACCA moves each patched fixture's levels from where they are to the cue's target. A recording has no targets and no fixtures — it is a stream of complete frames the console already shaped, replayed as-is. There is nothing for a curve to act on, so the tab can only mislead: before this change you could set a curve and a duration here, see no effect, and conclude the fade was broken. The transitions a recorded cue *does* have — how it enters on GO and how it leaves on Stop — are the **Fade in** / **Fade out** times on the Recording tab (see *Fade in and Fade out*).

Effect cues

An Effect cue runs a repeating pattern — a chase, a sine wave, a color cycle — *on top of* whatever your Light cues have set, without overwriting the underlying look. Create one with the **Effect** toolbar button. Its tabs are **Basics**, **Triggers**, and **Effect**.

[Illustration to be added in a future revision]

The **Effect** tab, top to bottom:

- **Preview** — a live waveform drawing one cycle of the configured pattern ("One cycle of the configured pattern. Changes as you edit.").
- **Type:** — the pattern: **Chase**, **Sine**, **Ramp**, **Pulse**, **Strobe**, **Random**, **ColorCycle**, **Rainbow**, **ColorChase**.

- **Target:** — which fixture parameter the effect modulates: **Intensity**, **Color**, **Pan**, **Tilt**.
- **Fixtures:** — which fixtures run the effect: **All Patched Fixtures** or any fixture group.
- **Presets:** — one-click starting points per pattern type ("Click a preset to populate every parameter below — you can still tweak after."). These are built-in quick-starts; there is no user preset save on this tab.
- Shape and spread: **Waveform Shape:** (**Sine**, **Ramp**, **Triangle**, **Square** — note that the Sine, Ramp, and Strobe types pick their own shape and ignore this), **Direction:** (**Forward**, **Reverse**, **Bounce**, **Random**), **Width / Duty:** (0–100%), **Phase Spread:** (0–360°), **Distribution:** (**Spread**, **None**, **Random**, **Pairs**, **OddsEvens**).
- Size and speed: **Amplitude:** (0–200%; 100% = full ±127 DMX), **Rate (Hz):** (1.0 Hz = one full cycle per second), or enter a **BPM:** — use **Tap Tempo** (tap 2+ times within 2 seconds to set the tempo to your average interval) and **Reset** to clear the tap history.
- **Colors** (for color effects): a **Mode:** choice of **Chase** / **Smooth Fade** / **Rainbow**; a swatch grid where each step color is a clickable swatch (click to open a color picker popup, **×** to remove), plus **+ Add Color**. The swatch list is hidden in Rainbow mode. A **Sequence preview:** strip shows the emitted gradient.
- **Fade In (s):** / **Fade Out (s):** — ramp the effect's amplitude in and out so a chase can build up and die away instead of snapping.

Fire the Effect cue with **GO** to start the pattern; stop the cue to end it. Because effects modulate on top of cue state, your base look returns untouched when the effect stops.

Recording DMX from a console

Console workflows vary enormously — busking desks, tracking desks, house boards with one good look. ATTACCA doesn't try to import your console's show file. Instead it does something simpler and more universal: **record what your desk outputs, and play it back as a cue.** Run your console's sequence once while ATTACCA captures the DMX; from then on, GO replays it exactly, no console needed.

> **Why it works this way:** a console show file is a proprietary description of *how* the desk arrives at its output. The DMX on the wire is the output itself, in one universal format every desk speaks. Recording the wire means any console works, with nothing to translate — at the cost that the recording is raw channel values with no idea which channel is a pan and which is an intensity. That trade shapes everything below, and the *Park cue* pattern is how you work with it.

Recording lives on the **Recording** tab of a Light cue.

[Illustration to be added in a future revision]

Signal Check before you record

Above the transport is a **Signal Check** section with a **▶ Check Signal** button. Press it before every recording session: ATTACCA listens with the *exact* receive path Record uses and shows, per universe, where the packets come from, how fast they arrive, and whether the channel values are moving. If nothing is arriving it says so — "No DMX arriving (universes 1..4). Check the console is outputting, and the interface in Preferences → Lighting." — and you fix the network *before* running the sequence, not after a take that turns out to be empty. Each

universe line also ends with **lost N** — packets the console numbered that never reached ATTACCA — and the block ends with a **Dropped:** total. Both should read 0; anything else means the wire, a switch, or this PC is losing data, and a take made now would be flagged (see *Loss is never silent*). It is safe to leave running while you record; ■ **Stop Monitor** ends it.

[Illustration to be added in a future revision]

> **Why it works this way:** the receive path has only two knobs that can be wrong — the universes and the network interface — but a wrong one fails silently on the wire (multicast simply never arrives). Signal Check exists so the failure is visible in five seconds instead of discovered after a ten-minute take.

Capturing a take

1. Point your console's DMX output at the ATTACCA machine's lighting network (sACN or Art-Net).
2. Select the Light cue and open its **Recording** tab.
3. Set the **Input Source**: **Protocol:** (**SacnMulticast** — the default — or **ArtNet**), **First universe:**, and **Universe count:** (up to 50 universes at once). The **Interface:** line is read-only and shows which network interface reception uses, "— set in Preferences → Lighting". Reception is *interface + universes*; the console's own IP address is never needed.
4. Run ▶ **Check Signal** and confirm every universe is arriving.
5. Click ● **Record**. Capture starts immediately — there is no separate arm step. The **Status** section shows live **State:**, **Duration:**, **Frames:**, **Data rate:** and **Dropped:** readouts. **Dropped** should stay at 0 for the whole take.
6. Run the sequence on your console.
7. Click ■ **Stop**. Click ✕ **Clear** instead if you want to discard the take and go again.

> **Why the interface lives in Preferences:** the receive interface is the same NIC that carries ATTACCA's own sACN/Art-Net output, and it has one home — Preferences → Lighting → Network Interface. The Recording tab used to have its own "Network IP" field; operators typed the *console's* address into it (a natural guess), which is meaningless for multicast reception and made recordings fail silently. One setting, in one place, shown read-only where you need to see it.

The first frame waits for every universe. When Record starts, ATTACCA holds the first frame for up to 250 ms until every configured universe has delivered at least one packet, so frame 1 is a complete snapshot of the rig rather than whichever universe happened to arrive first. A universe that is still silent when the warm-up ends is recorded as zeros from that point until it transmits, and the log notes which universes were "still silent" at record start — if you see that line and didn't expect it, the console wasn't outputting that universe yet.

> **Why it works this way:** without the warm-up, the first recorded frame captured only the first universe to arrive and zeros for the rest — and on playback, GO would snap every other universe to zero for one frame before the real data landed. The rig lurched on every fire. 250 ms is longer than any console's frame interval and short enough to be invisible in a take.

Two automatic protections:

- **Capture ceiling.** Recording auto-stops at **200,000 frames or 30 minutes**, whichever comes first — and the take is **kept**, exactly as if you had pressed Stop. The status reads "Auto-stopped at 30-min limit — saved N frames."
- **Saving in the background.** After Stop, the status shows **"Saving recording…"** — long takes can take a moment to write, but the UI stays responsive; Record and Clear are disabled until the save finishes ("Stopped — saved N frames").

Every console frame is captured. Capture is driven by the console's packets, not by a timer of ATTACCA's own: each packet that arrives, for each universe, is stored in exactly one frame, in order, up to the maximum rate sACN allows (44 per second per universe). A frame is one console cycle — the packets of all your universes that belong together — and it is stamped with the moment its first packet arrived, so playback reproduces the console's own timing. Universes the console did not send in a cycle keep their previous values in that frame. The **Frames:** count is therefore the number of console cycles in which *something* was sent: a console that only transmits universes that changed (grandMA3 does this) produces fewer frames on a static look than on a chase, and the **Data rate:** readout — frames ÷ duration — says what was actually captured, which is different from the console's cycle rate and is supposed to be.

> **Why it works this way:** a strobe is a one-frame event. The first build captured one snapshot per console cycle, taken at the moment the cycle's *first* packet arrived — every universe whose packet came a few microseconds later was stored a cycle late, and when the console varied the order between cycles, a universe's frame was overwritten before it was ever stored. On a smooth chase that is invisible (the developer's clean-looking takes had already lost about one in nine frames of universes 1 and 2); on a flash it is a missing light and a stagger. Storing what arrives, when it arrives, is the only design that never does worse than the console.

What capture can and cannot represent. ATTACCA never captures fewer frames than the console sends. A strobe the console can express — one cycle on, one cycle off, i.e. up to half the console's output rate (15 flashes a second on a 30 Hz desk) — is captured edge for edge. Anything faster than half the console's rate cannot exist in the console's own DMX stream, so it is the console's limit, not the recording's; for faster strobes use the fixture's strobe channel, which is a static value and records perfectly.

How much it can take. Measured on an ordinary Windows 11 PC: **50 universes at 44 packets a second each — the most sACN allows, at ATTACCA's universe cap — with every channel of every universe changing every cycle, captured with zero loss at about a fifth of one processor core**; a five-minute run at 16 universes lost nothing. That is the honest ceiling of the input side, and it is above anything a console can send. Memory grows only with what changes: a take in which everything changes every frame costs about 21 MB per minute at 16 universes; a chase or a static look far less. If a machine ever falls short of this, the **Dropped** counter says so — nothing is lost quietly.

Loss is never silent

Every sACN and Art-Net packet carries a sequence number, so ATTACCA knows when one the console sent never arrived (a switch, a bad cable, a saturated NIC) and when one arrived but could not be handled in time (the PC itself). The two are counted separately and shown together:

- while recording, the Status block's **Dropped:** line counts them live and reads *"N — take will be flagged"* the moment the first one is lost;
- after Stop, the **Attached Recording** block reads `Frames: N · Data rate: X fps` and either `Dropped: 0 (P packets, every one captured)` or `Dropped: N packet(s) — NOT a faithful capture; re-record if it matters`; the log carries a warning naming the universes;
- a take recorded by an earlier build reads `Dropped: unknown (recorded before loss counting)` — nothing was counted then, so nothing is claimed.

A flagged take still plays; what it is missing is the console frames that were lost, which cannot be recovered after the fact. If the number matters — a strobe, a fast chase — fix the cause (run ▶ **Check Signal** and watch its **lost** column while the console runs) and record again. Zero is the normal reading.

[Illustration to be added in a future revision]

> **Why it works this way:** a lost packet used to be invisible: the take simply had one frame fewer and nobody could tell whether the console, the network or ATTACCA had dropped it. A number you can see, per universe, split by where the loss happened, turns "the strobe looked wrong" into a fact you can act on before the show.

Where recordings are saved

The take is written as a `.dmxrec` file in a `recordings` **folder next to your workspace file** — for a workspace on the Desktop, `Desktop\recordings\cue_3_<timestamp>.dmxrec`. It is *not* stored in AppData or anywhere else on the machine.

> **Why it works this way:** the recording is part of the show. Keeping it beside the `.atca` file means copying the workspace folder to the show laptop copies the takes with it, and a workspace that opens on another machine plays exactly what was recorded. A take hidden in a per-user profile folder would be the one thing that didn't travel.

On load, `.dmxrec` files larger than 1 GiB are rejected as a safety cap; in practice that only matters for very long multi-universe recordings, in which case split the material across shorter takes (a cue whose recording fails to load falls back to its normal levels/commands).

Playing it back

Nothing special: a Light cue with an attached recording plays the recorded frames instead of its levels when you press **GO**. The first recorded look is emitted immediately on fire, and every frame is written to each universe as one whole frame. A non-looping take completes when its last frame has played; by default the last frame's values stay on the rig until something else writes them — the **On finish** setting can change that (see *What happens when a recorded cue finishes*).

> **Why it works this way:** a recorded frame is one snapshot of the rig. Writing it as 512 separate channel updates meant the output could be sent half-way through the update — a moving light's 16-bit pan split across two frames — which is what made fixtures twitch toward the wrong direction during playback. Whole-frame writes mean the wire only ever carries a frame that was actually recorded.

Reopening a saved workspace. A cue's take is read from its `.dmxrec` file the first time it is needed in a session — when you select the cue (the Timeline shows *Loading take from disk…* for a moment, then the activity graph and the Analysis block with the stored loop seam) or when you GO it, whichever comes first. It is read once; every later selection and GO uses the frames already in memory. If the file cannot be read the Analysis block says why (*Could not load the take:* …) and the cue falls back to its plain levels.

> **Why it works this way:** the show file holds only the take's header (its length, trim and loop points) so it stays small; the frames live next to it. Reading them on selection means the tab always shows the take you are about to edit, not an empty timeline that only fills after the first GO — which is what happened before, and looked like the recording had gone missing.

What the Duration column shows. For a recorded cue, the cue list's **Duration** is the length of the part of the take that will play — the trimmed region — formatted like every other cue (`00:15.90`). When the cue loops, it shows one pass with an infinity mark: `∞ 00:15.90` . While the cue runs, the column counts down the current pass (`∞ −00:12.3`) and the Active Cues panel shows elapsed / remaining the same way; on a loop the countdown restarts each pass, like a looping audio cue. Trimming or changing the loop on the Recording tab updates the column at once. Fixture-based Light cues are unchanged: their Duration is the fade time.

> **Why it works this way:** the take *is* the cue's timeline, so that is what the column should report — the fixture fade time on the Curve tab has nothing to do with recorded playback (see *Fade times (the Curve tab)*). The ∞ mark exists because a looping take never ends on its own: it runs until the next recorded cue takes its universes, a Park cue, Stop or Panic. Showing one pass tells you how long a cycle is; the mark tells you not to wait for it to finish. Auto-follow on a looping recorded cue fires after one pass, exactly as it does for a looping audio cue. The Duration box on the Basics tab shows the take's length but cannot change it — the length belongs to the take; edit the trim on the Recording tab.

Cue priority: the newest recorded cue takes the universe

A recorded light cue that fires takes priority. If another recorded cue is still running on any of the same universes, it is stopped the instant the new one fires — silently: no blackout frame, no flash, no mover swing — and the new cue's frames take over. The stopped cue shows as ended in the cue list and will not come back (not after a Panic, not on the next GO). Recorded cues on *different* universes are unaffected and keep running side by side. This is what lets you GO straight from a chase into a movement cue, or from anything into the Park cue, without stopping anything first.

> **Why it works this way:** a recorded take is a *complete* picture of its universes, 44 times a second. Two complete pictures of the same universe can't be merged the way two programmed cues can (there is no "highest takes precedence" between whole frames, and ATTACCA doesn't know which channel is an intensity) — so when two ran at once, the rig showed whichever frame arrived last, alternating tens of times a second: a flicker. The only honest rule is that one cue owns a universe at a time, and the one you fired most recently is the one you meant. The handoff happens in a single step, so the wire never carries one frame of the old cue after the first frame of the new one.

Fade in and Fade out

The **Playback** section of the Recording tab has two per-cue times, both in **seconds** and both **0 by default** (an instant cut, like a console). They are typed and shown exactly like **Duration**, **Pre Wait** and **Post Wait** on the Basics tab: `2`, `1.5` or `0:02.50` all work, the box commits on Enter or when you click away, and an empty box means 0.

- **Fade in (seconds)** — on **GO**, including when this cue takes over from another recorded cue, ATTACCA crossfades from *whatever is on the wire right now* (the outgoing cue's current frame, a programmed look, or darkness) to this take's **live** frames over the fade time: the take starts playing at GO and its opacity ramps up over it, exactly as an audio or video fade does. A colour scroll scrolls while it fades in; movers glide from where they were toward where the take *is* at each moment. The take's clock runs from GO, so a 2-second fade over a 15-second take is 15 seconds on stage, and the Duration countdown starts at GO. A fade longer than the take holds the take's last frame under the rest of the ramp, then finishes.
- **Fade out (seconds)** — on a manual **Stop** (the Stop key, the transport button, or OSC), ATTACCA dims the take to black over the fade time while it keeps playing, then holds black. A fade that outlasts the take keeps dimming the last frame and ends in zeros. **Panic is not affected** — it keeps the panic fade time set in General preferences.

[Illustration to be added in a future revision]

> **Why seconds:** every other timing field in ATTACCA is in seconds, and a fade you can see is measured in seconds. The first build of this feature took milliseconds, and "2" typed into it produced a 2 ms fade — a snap with a fade nobody could see. A show file saved by that build is read correctly: a stored value of 10 or less is taken as the seconds you meant, anything larger as the milliseconds it said it was.

> **Why the content keeps moving:** a fade is an opacity ramp, not a pause. The first build held the take on its first frame until the ramp landed, so a two-second fade in front of a colour scroll showed a frozen colour for two seconds and then jumped into motion — which is not what a fade means anywhere else in ATTACCA (an audio fade-in does not delay the music). Playing under the ramp is also what makes a Fade in on a chase look like the chase arriving, rather than a still photograph of it.

> **Why they are separate from Duration:** for a fixture-based Light cue, **Duration** *is* the fade time — the cue has nothing else to time. A recorded cue is different: the take has its own timeline (the Duration column shows it; see *Playing it back*), and the fade in is something that happens *in front of* that timeline, the fade out something that happens *after* you stop it. Two extra numbers, not one overloaded one.

> **Why it works this way — and the honest limit:** the fade is a crossfade of raw DMX values, because that is all a recording is. It fades *every* channel, pan and tilt included. So a fade *in* from a parked look glides the movers to position (good), but a fade *out* to black sweeps them toward their zero position while dimming (usually not what you want). For controlled darkness, GO your Park cue — the newest-cue rule above hands over to it cleanly, and Fade in on the Park cue gives you a smooth glide home. Fade out exists for rigs without movers, or for the cases where "everything to zero, gently" is exactly right.

Checking a fade in the log. When a recorded cue fires, ATTACCA writes a line like `RecordedLightCueAction: started — 549 frames, 2 universes, loop=True mode=AsRecorded region 0→548 crossfade=0ms fadeIn=2s fadeOut=2s` to the log in `%APPDATA%\ATTACCA\logs`. The last two values are what the cue handed over on that GO — if you see `fadeIn=0s` the field is empty on *that* cue.

The Curve tab does not apply to recorded cues. With a recording attached, the Curve tab's controls are greyed out behind a one-line note. See *Fade times (the Curve tab)* for why.

> **One more caveat — 16-bit channels during a blend.** A crossfade blends each DMX byte on its own. A moving light's 16-bit pan or tilt is two bytes (coarse and fine) and ATTACCA can't tell which two belong together, so during a blend the fine byte can be off by up to one coarse step — a slight wobble on a slow glide, never a jump to a position the take didn't contain. Loop crossfades have the same limit. The patch-aware recording feature planned for v1.1 fixes it by knowing which channels are 16-bit pairs.

The Playback output line

Under **Interface:** on the Recording tab is a read-only **Playback output:** line — for example `sACN multicast — Ethernet 10.101.10.5 — set in Preferences → Lighting`. It shows what *this computer* will transmit when the cue plays, exactly as configured in Preferences → Lighting, and it updates when you save Preferences. It is not a setting.

> **Why it works this way:** output settings have one home (see *Enabling lighting output*), so the tab shows them rather than offering a second place to change them — two places to set the same thing is how the network interface ended up wrong before. If the line says "none enabled", nothing will reach the rig: open Preferences → Lighting and enable your protocol.

Loop modes

The **Loop mode:** selector on the Recording tab offers three modes:

- **Off** — play the take once.
- **Loop as recorded** — repeat the whole trimmed take, end to start, with the crossfade at the seam. Always available; use it when the take was recorded to loop (a chase, a color cycle) and you want exactly what the

desk did.

- **Auto-trim loop (ATTACCA-sensed)** — repeat between loop points ATTACCA found by analysing the take for the moment the sequence returns to its starting look. The **Detected loop:** readout shows the points and the detector's confidence — "Detected: 00:00.00 → 00:12.48 (frames 0→549), confidence 99% — applied". At **60% or higher** the detected loop is applied automatically and the cue switches to Auto-trim; below that it reads "below 60% — manual review recommended" and an **Apply detected loop** button lets you use it anyway after you've watched the take.

Crossfade (ms): blends the loop-end frame into the loop-start region across the seam (default 500 ms; 0 = snap cut). It is per cue.

> **Tip — crossfade length by mode.** With **Auto-trim**, keep the crossfade **short (0–100 ms)**: the seam is already matched by the detector, so there is almost nothing to blend, and a 500 ms crossfade reads as a brief *pause* at the loop point while two nearly identical looks fade into each other. With **Loop as recorded**, a longer crossfade is what hides a seam the desk never intended to loop.

- **Trim:** the **Timeline** control shows the take with draggable trim and loop handles — drag to cut dead air off the ends or tighten the loop points. **Reset** clears trim/loop and uses the full recording; **Re-analyze** re-runs the loop/trim detection.
- The **Attached Recording** info block shows the take's metadata.

> **Why two loop modes:** the detector is guessing where a sequence repeats from raw channel motion — usually right, occasionally wrong on a take with a long static section. "Loop as recorded" is the guess-free fallback that always does the obvious thing, so a wrong guess never costs you a working loop.

Stop, Panic, and the Park cue

Stop and Panic send every recorded channel to zero. Stopping a recorded cue writes a zero frame to each of its universes (after the cue's **Fade out** time, if you set one); a single panic fades all DMX to black over the panic fade time, and a double panic zeros everything immediately and holds it there (see *Grand master, blackout, and panic*). On a rig with moving lights that means **Stop swings the movers to their zero position** — pan and tilt to one extreme, at full speed with no fade, or as a sweep with one.

> **Why it works this way:** a raw DMX recording has no fixture knowledge. ATTACCA cannot tell that channel 17 is a pan and channel 20 is an intensity, so it has no way to "fade only the intensity" or "hold position" — the only honest stop it can perform is *everything to zero*. Panic is the emergency stop and everything-off is exactly what it should mean. For a *planned* stop, use a Park cue.

The Park (Standby) cue pattern — recommended. Record a short take of the rig in its parked-dark state: movers at a safe home position, intensity at 0, no movement — a few seconds is plenty. Give it a name like "PARK" and place it wherever the recorded sequence should end. **GO** the Park cue to bring the rig to controlled darkness: the recorded frames put each fixture exactly where you parked it, at zero intensity, instead of slamming to an extreme. Fire the next recorded cue from there; leave Stop and Panic for when something has gone wrong.

You never need to stop a recorded cue before firing the next one. The newest recorded cue takes over any universes it shares with a running one (see *Cue priority*), so the normal way to end a sequence is simply to GO the next cue — and the Park cue with a short **Fade in** is the graceful way to end it in darkness.

What happens when a recorded cue finishes

A take with **Loop mode: Off** plays once and then *finishes*. The **On finish** section at the bottom of the Recording tab says what the rig does at that moment. It has three settings:

- **Hold last frame** (the default) — nothing. The final look of the take stays on the wire until another cue writes those universes. Nothing moves, nothing goes dark.
- **Blackout** — ATTACCA writes one blackout frame to the take's universes: every channel to zero. It is exactly the frame Stop writes, so it has exactly Stop's side effect: on a rig with moving lights, **pan and tilt go to zero too and the movers swing** to their end stop. Use it on rigs without movers, or when "everything off, now" is genuinely what the moment needs.
- **Go to cue → [a Light cue]** — ATTACCA fires the chosen cue the instant the take finishes, *exactly as if you had pressed GO on it*: it takes over the universes (see *Cue priority*), and its own **Fade in** blends from the finished look to its first frame. The picker offers every Light cue in the show except the cue itself.

[Illustration to be added in a future revision]

On finish only fires when a non-looping take reaches its end on its own. It never fires when you **Stop** the cue, when you **Panic**, or when another recorded cue **takes over** its universes mid-take — those are interruptions, not a finish. And a looping cue never finishes at all, so while a loop mode is selected the control is greyed out with the note *Looping cues don't finish — use a GO to move on.*

> **Why Hold is the default:** a raw DMX recording has no fixture knowledge, so the only things ATTACCA can do at the end of a take are "leave it" or "zero it" — and zeroing swings movers (*Stop, Panic, and the Park cue*). The one action that cannot move anything on any rig is doing nothing. So finishing does nothing unless you ask for something, and when you ask for Blackout the tab's tooltip tells you what it costs.

> **Why Go to cue goes through the normal GO path:** the target is fired the same way a keypress fires it, not by some private shortcut. That is what makes it inherit everything a GO already has — the target's Fade in, the newest-cue-wins takeover, the arming and license checks, the auto-follow rules of the target's own list. It also means a chain of finishes (A finishes → fires B, B finishes → fires A) is just a playlist that plays itself: each hop happens on the engine's clock when a take actually ends, never as one cue re-entering another, so the safety guard against cue chains that loop back on themselves (see *Start* in *Cue Types: Control*) is never involved. A cue is not allowed to go to *itself*, because "play, then fade back into my own start" is what a loop with a crossfade already does, and two ways to say the same thing is how loops get set twice by accident.

Is it an auto-follow? No — and the two are easy to confuse. **Auto-follow** on the *next* cue in the list ("start me when the previous cue ends") is ATTACCA's general sequencing rule for every cue type, and on a *looping* recorded cue it fires after **one pass**, exactly as it does after one pass of a looping audio file. **On finish** belongs

to the recorded cue itself, points at *any* Light cue in the show (not just the next one), and fires only when a *non-looping* take completes. Use auto-follow to sequence a list; use On finish to return a one-shot look to a standing one.

A dangling target. If the cue you chose is later deleted, the take finishes with **Hold** instead, the log notes it once, and the Recording tab shows *Target cue not found* until you pick another.

In the log. When a take finishes, ATTACCA writes one of `RecordedLightCueAction: finished — onFinish=Hold`, `… onFinish=Blackout`, or `… onFinish=GoToCue target="STANDBY"` to the log in `%APPDATA%\ATTACCA\logs`, followed by the target's own start line if there was one.

Worked example — a chase that returns to Standby. This is the pattern On finish exists for.

1. **STANDBY** is a recorded Light cue: a few seconds of the rig parked (movers home, intensity 0), **Loop mode: Loop as recorded**, **Fade in: 2**. It runs all evening; nobody presses Stop on it.
2. **CHASE** is a recorded color chase, **Loop mode: Off**, **Fade in: 1**. On its Recording tab set **On finish: Go to cue → STANDBY**.
3. GO **STANDBY**. The rig sits parked.
4. GO **CHASE**. It takes the universes from STANDBY in one step (no flicker), blends in over 1 second, and plays through once.
5. When the last frame has played, CHASE finishes and fires STANDBY — a normal GO. STANDBY takes the universes back and its 2-second **Fade in** glides the movers from the chase's final positions back to home while the intensity comes down to 0. The log reads `finished — onFinish=GoToCue target="STANDBY"`, then STANDBY's start line.
6. The next GO in the list — cue 3, cue 4 — takes the universes from STANDBY exactly as CHASE did. Nothing ever needed a Stop, and the rig never saw a frame the console did not produce, apart from the two glides you asked for.

Set **On finish: Blackout** on CHASE instead and step 5 becomes a hard cut to zeros — colors off and movers swinging home the fast way. That is the honest raw-DMX behaviour, and it is why the default is Hold and the recommended return is Go to cue.

The recommended recorded-cue workflow

Putting the pieces together, this is the shape of a recorded lighting sequence that behaves on stage:

1. **Start with a Standby (Park) take.** Record a few seconds of the rig parked: movers home, intensity 0. Name it "STANDBY" or "PARK", set it to loop, give it a Fade in. It is both the first cue and the last.
2. **Give the movement cues a Fade in.** A take that starts with the movers somewhere else glides there over the fade instead of snapping (2 seconds is a good first value); colors blend the same way. Set it per cue on the Recording tab, in seconds.
3. **GO cues over each other freely.** Fire the next recorded cue whenever the show needs it — no Stop in between. The newest cue takes the universes in a single step: no flicker, no blackout frame, and the Fade in on the incoming cue blends from whatever the outgoing cue was showing.
4. **Let one-shot cues return to Standby by themselves.** On every non-looping take set **On finish: Go to cue → STANDBY**. When the take ends the rig glides back to the parked look through Standby's Fade in, and the next GO takes over from there (*What happens when a recorded cue finishes*).

5. **End in the Standby take.** GO the Park cue for controlled darkness. Its recorded frames put every fixture exactly where you parked it, at zero intensity; a Fade in on it glides the movers home.
6. **Stop and Panic zero everything.** Both send every recorded channel to zero — pan and tilt included, so movers swing to an extreme. Keep them for when something has gone wrong, not for ending a sequence; a Fade out on Stop softens the dim but not the swing.

[Illustration to be added in a future revision]

> **Why this order:** each step leans on one rule above. The Park take exists because ATTACCA cannot fade "only the intensity" of a raw recording (*Stop, Panic, and the Park cue*). Fade in is the only transition a take has (*Fade in and Fade out*). GO-over-GO works because one cue owns a universe at a time and the newest wins (*Cue priority*). On finish → Go to cue is a GO the take presses for you, so it obeys the same two rules. Follow the shape and the rig never sees a frame the console did not produce, except the blends you asked for.

The Light Dashboard

View → Light Dashboard… (Pro) opens a live lighting console panel — for focus sessions, quick looks, busking, and grabbing states into cues.

[Illustration to be added in a future revision]

What's in it, top to bottom:

- **Tabs and views: Live** (writes to the rig) vs **Audition** (a blind scratchpad — build a look without outputting it, with **Reset from Live** to copy the current state in). View modes: **Grid** (16-column channel cells showing fixture name, channel number, editable value, and function; parked cells are striped), **Faders** (a **GM** grand-master fader plus per-channel vertical faders, parked channels badged **P**), and **Fixtures** (a fixture control panel with selection, **INTENSITY** (Dimmer/Strobe), **COLOR** sliders, **POSITION** (Pan/Tilt), **BEAM** (Gobo/Focus/Zoom/Iris/Prism), and other channels). The right side shows **Latest cue:** — the most recent light cue that fired.
- **Command line:** type lighting commands (e.g., `1 = 100`, `backlight = 50`) and press **Enter**. The **Channels:** box plus **Over Time:** (seconds) fades specific channels over time (empty = all channels).
- **Controls row: Universe:** selector, channel filter **All** / **Used**, the **Grand Master:** slider (0–100), and **BLACKOUT** (Live only), **Revert**, **Clear All** (Live only), **Reset from Live** (Audition only).
- **Color Picker panel** — toggled by the **Color…** button ("Show the live color picker — pick a color and the selected fixtures change immediately."). It sends **live DMX**: a **Group:** dropdown with **All**/**None** and a fixture checklist, the color wheel, quick swatches (**Red**/**Green**/**Blue**/**White**/**Amber**/**Off**), and the **Gels (Rosco / Lee)** expander. The banner reminds you of its scope: **LIVE — dashboard color overrides until next cue fires.**
- **SUBCONTROLLERS** — a fader strip of every Light cue that has **Use as subcontroller in dashboard** checked, so you can ride cue looks like submasters.
- **Park / Record row: Park** holds a channel at its current value regardless of cues ("Park the right-clicked channel at its current value") and **Unpark** releases it — right-click a channel cell first to choose the target

(there's no menu; the right-click silently selects the cell). **Record All…**, **New Cue with All**, and **New Cue with Changes** capture the current levels into new Light cues — the fastest way to turn a busked look into programming.

The DMX Status window

View → DMX Status… opens a diagnostic view of what ATTACCA is actually putting on the wire — your first stop when a fixture isn't responding. (Unlike the Dashboard, it is not tier-gated.)

It shows one universe at a time: a **Universe:** selector, a legend (**Off** / **Active** / **Parked**), and a 512-channel grid (16×32) of channel-number/value cells, color-coded and refreshed continuously. Cells are directly editable — typing a value writes that channel as a manual override, handy for "is it the cue or the cable?" tests — and **Clear Overrides** removes them all. The status bar summarizes: universe number, channel count, how many channels are active and parked.

[Illustration to be added in a future revision]

Grand master, blackout, and panic

Three different "make it dark" controls, with three different behaviors:

- **Grand Master** (Dashboard toolbar slider, and the **GM** fader in Fader view) scales **all** DMX output proportionally at the output stage — 50% halves every channel on every universe. It's a volume knob for the whole rig.
- **BLACKOUT** (Dashboard, Live tab) is a **dead blackout (DBO)**: output is suppressed but the look is preserved underneath. Release it and everything restores instantly — exactly what you want for a planned blackout you'll come straight back from.
- **Panic** (`Esc`) is the emergency stop: a single panic fades all DMX to black over the panic fade duration; a second `Esc` within one second **zeros every channel in every universe immediately** and holds them at zero until the next cue writes something. Unlike DBO, panic doesn't preserve the look — it's the "something is wrong" button, and it stops your audio and video too (see *Running the Show*).

> **Why it works this way:** panic is deliberately the one control that never asks what the channels mean. A recorded take or a running effect might be driving pan, tilt, strobe, or intensity — ATTACCA can't tell for raw DMX — so the only stop that is *guaranteed* safe is every channel to zero, held there until a cue you fired on purpose writes something new. Everything-off is the point. For a planned, graceful dark, use a Fade cue on programmed looks or a recorded Park cue on console takes.

Worked example: four LED pars, a warm wash, a chase, and a console take

A complete first lighting session, assuming four RGBA LED pars on universe 1, addresses 1–32 (8 channels each), connected through an sACN node.

1. Enable output.

1. **Edit → Preferences… → Lighting**.
2. Under **Network Interface**, pick the adapter wired to your lighting network.
3. Check **Enable sACN output**. Leave **Multicast mode** on, Priority at 100.
4. Under **DMX Universes**, leave **Universe Count** and **Start Universe** at 1. Click **Save**.

2. Patch the pars.

1. **View → Light Patch… → Patch Fixtures…**.
2. Click **Browse…**, search the Fixture Library for your par model, pick the 8-channel mode, and click **Select**.
3. Set **Quantity** 4, **Universe** 1, **Start Addr** 1, **Group Name** `wash`, **Naming** to **Named (Fixture 1, Fixture 2…)**. Check the address preview reads 1, 9, 17, 25 — then click **Apply**.

3. Build the warm wash.

1. Click the **Light** toolbar button (`Ctrl+6`); name the cue "Warm wash" in **Basics**.
2. On the **Levels** tab, in the Color Picker section choose the `wash` group and click **All**.
3. Tick **LIVE EDIT** so you can see the rig respond, then click the **Warm White** quick swatch (or open **Gels (Rosco / Lee)** and **Apply** a favorite gel) and set **Intensity:** to taste.
4. Click **Save current…** in **My presets** to keep the color for later. Untick **LIVE EDIT**.
5. On the **Curve** tab, set **Duration (seconds):** to 3 for a gentle fade-up.

4. Add a color chase on top.

1. Click the **Effect** toolbar button; name it "Sunset chase".
2. On the **Effect** tab: **Type: ColorChase**, **Target: Color**, **Fixtures:** `wash`.
3. In **Colors**, keep **Mode: Chase** and use **+ Add Color** to add amber, red, and magenta step swatches.
4. Set **Rate (Hz):** around 0.5 — or tap **Tap Tempo** in time with your music.
5. Set **Fade In (s):** 2 so the chase builds in. Fire the Light cue, then the Effect cue: the chase runs over the warm wash, and stopping the Effect cue returns the plain wash.

5. Record a take from the house console.

1. Have the console output sACN universe 1 onto the same network.
2. Create a new Light cue, open its **Recording** tab: **Protocol: SacnMulticast**, **First universe:** 1, **Universe count:** 1. Check the **Interface:** line names your lighting NIC (set in Preferences → Lighting), then press ▶ **Check Signal** and confirm universe 1 is arriving.
3. Click ● **Record**, run the console sequence, click ■ **Stop**, and wait for "Saving recording…" to finish.
4. Drag the **Timeline** trim handles to cut the dead air. If it should repeat, set **Loop mode:** to **Loop as recorded** (or keep the **Auto-trim loop** ATTACCA applied if the detected loop reads ≥ 60%, and shorten the crossfade to 0–100 ms), then press **GO** — the desk's sequence is now a cue.
5. Record one more short take with the console in its parked-dark look and name it "PARK": that is the cue you fire to end the sequence cleanly, because Stop would zero every channel and swing any movers to an extreme. Give PARK a **Fade in** of a second or two so the movers glide home instead of snapping. You don't need to stop the running take first — the newest recorded cue takes over the universes on GO.

Common issues

- **Fixtures don't respond at all** → Check Preferences → Lighting: is the right protocol enabled, and is the **Interface** set to the adapter actually wired to your rig? On multi-adapter machines, "all interfaces" often means the packets leave the wrong port. Then open **View → DMX Status…** — if channels show Active there, ATTACCA is outputting and the problem is downstream (node config, universe number, cabling).
- **Wrong fixture moves when you program** → Address or universe mismatch. Compare the fixture's physical address/mode against its row in the Light Patch grid; remember ATTACCA universes are 1-based while the **Art-Net Universe** field is 0-indexed per that protocol.
- **Channel labels in the Levels grid look stale after a patch change** → Labels refresh when the tab binds or the universe switches. Switch universes and back, or reselect the cue.
- **Deleted the wrong instrument or group** → Light Patch deletes are immediate with no confirmation. Re-patch it (the Patch Fixtures dialog makes this quick) — and select carefully before **Delete Selected**.
- **The rig follows your edits while programming** → **LIVE EDIT** is on in the Levels tab (look for the red ● "(live to wire)"), or the Dashboard color picker is open on the **Live** tab. Untick LIVE EDIT, or build in the Dashboard's **Audition** tab instead.
- **Everything is dark but cues are "running"** → Check the Dashboard's **Grand Master** (is it pulled down?), **BLACKOUT** (is DBO engaged?), and whether a double-`Esc` panic zeroed the rig — after a panic blackout, channels stay at zero until the next cue writes them.
- **A dashboard color override won't go away** → Dashboard color picks are live overrides that last "until next cue fires." Fire any Light cue, or use **Clear All** / **Revert** on the Live tab; parked channels need **Unpark**.
- **A recording won't attach or plays the cue's plain levels instead** → The `.dmxrec` file may be missing (keep the `recordings` folder next to the workspace) or over the 1 GiB load cap. Re-record in shorter takes; a rejected recording makes the cue fall back to its normal light action.
- **The console take recorded nothing (0 frames)** → The console must be sending to the network interface ATTACCA is listening on. Run ▶ **Check Signal** on the Recording tab: if it reports "No DMX arriving", match the **Protocol** and **First universe** to the desk's output and set the receive NIC in **Preferences → Lighting → Network Interface** (the **Interface:** line on the tab shows which one is in use).
- **One universe plays back as all zeros for the first moment, or is dark for the whole take** → Check the log line written at Record start: universes "still silent" after the 250 ms warm-up are recorded as zeros until they transmit. If the console never sent that universe, the take has no data for it — re-record with the desk already outputting.
- **Stopping a recorded cue swung the movers to an extreme** → Expected: raw DMX has no fixture knowledge, so Stop and Panic zero every channel. End recorded sequences with a recorded **Park** cue instead (see *Stop, Panic, and the Park cue*).
- **The Recording tab says "Loading take from disk…" or the timeline is empty after reopening a show** → The frames are being read from the `.dmxrec` file; they appear in a moment. If the Analysis block says *Could not load the take*, the file is missing or unreadable — keep the `recordings` folder next to the workspace.
- **A strobe or fast chase plays back with lights missing or out of step** → Look at the take's **Dropped:** line in the Attached Recording block. If it is above zero, packets were lost on the way in (run ▶ **Check Signal** and watch the **lost** column while the console runs to find where — a switch, a cable, or this PC) and the take must be re-recorded; a take that reads "unknown" was made by a build that sampled the console

instead of capturing every packet — re-record it with this version. If Dropped is 0, the take is faithful: compare the flash rate with half the console's output rate (see *What capture can and cannot represent*).

- **A recorded cue stopped by itself when I fired another one** → Expected: the newest recorded cue takes over any universe the two share (see *Cue priority*). Put cues that must run together on different universes.
- **Nothing transmits after updating ATTACCA, but it did before** → Output settings moved from the show file to this computer's Preferences. Open **Preferences** → **Lighting**, enable your protocol, pick the interface, and click **Save** once; the Recording tab's **Playback output:** line confirms what will transmit.
- **Opening a workspace changed my lighting output** → It can't any more: Preferences → Lighting is the only place output is configured. If the **Playback output:** line doesn't match what you expect, the setting is in Preferences, not in the file.
- **A fade out swung the movers** → Expected: raw DMX fades every channel, pan/tilt included. Use Fade out only on rigs without movers; end sequences with a Park cue (with Fade in) instead.
- **Effect cue seems to ignore its Waveform Shape** → By design: the **Sine**, **Ramp**, and **Strobe** types pick their own shape; Waveform Shape applies to the other pattern types.
- **Other sACN software on the same machine can't see ATTACCA's output** → Enable **Multicast loopback (same-machine monitoring)** in Preferences → Lighting (it's on by default; turn it back off for production).

Control & Integration

ATTACCA is built to sit at the center of a show network: it can take orders from a lighting console, a stage manager's phone, or a MIDI keyboard — and give orders right back. Most of this chapter — MIDI, OSC, Lua scripting, collaboration, and broadcast — is part of **ATTACCA Pro**. The **Web Remote** is the exception: it is included in **Standard** as well.

MIDI

Enabling MIDI devices

ATTACCA never opens a MIDI port you haven't asked for. Every device is opt-in:

1. Open **Edit → Preferences...** (`Ctrl+,`) and select the **MIDI** page.
2. Make sure **Enable MIDI** is checked (it is by default).
3. Under **MIDI Devices**, find your device in the **Input Devices** or **Output Devices** list and check its box. All devices start unchecked — nothing is enabled until you do this.
4. Confirm **Enable MIDI input (receive)** and/or **Enable MIDI output (send)** are checked for the direction you need.
5. Click **Save**.

[Illustration to be added in a future revision]

Two more settings on this page matter for triggering:

- **MIDI Channel** (under "Workspace Channel") — `0` means listen on any channel; `1` – `16` restricts workspace-level MIDI to one channel.
- **Middle C** (under "Note Display") — choose **C4 (Traktor, Resolume)** or **C3 (Ableton, Logic)** so note names in ATTACCA match the labels in your other gear. This only changes how notes are displayed, never how they're matched.

MIDI mapping profiles

The **Mapping Profiles** section at the top of the MIDI page lets you save the page's settings and workspace control mappings under a name and recall them later — handy when you tour between venues with different hardware. Use **Save Profile** to store the current setup, **Load** to apply a saved one, **Delete** to remove it, and **Export** / **Import** to move profiles between machines as `.atca-midi-profile` files. Profiles are stored per-user in `%APPDATA%\ATTACCA\midi-profiles` .

Triggering a cue from MIDI (MIDI Learn)

Any cue can be fired by an incoming MIDI message. The fastest way to set one up is to let ATTACCA listen:

1. Enable a MIDI input device (previous section) — the **Learn** button stays disabled until at least one input device is on.

2. Select the cue and open the inspector's **Triggers** tab.
3. In the **Trigger Sources** column, check **MIDI Trigger**.
4. Click **Learn**. The button changes to **Listening…**.
5. Within 10 seconds, press the key, pad, or fader move you want on your controller. If nothing arrives in 10 seconds, the capture times out and the button returns to **Learn** — just click it again.
6. The captured message fills in the fields below: **Type** (the message type), **Ch** (channel), **Note** (the note or controller number), and **Vel** (the value rule).

[Illustration to be added in a future revision]

You can also fill the fields in by hand, and the **Copy** / **Paste** buttons move a complete trigger setup from one cue to another.

Two things to remember: the cue must be **armed** for its MIDI trigger to fire, and a channel of `0` means "follow the workspace MIDI Channel" from Preferences.

Trigger types and the value rule

Type	Fires when ATTACCA receives…	**Vel** field
NoteOn	a note press	Matches the velocity
NoteOff	a note release	Matches the release velocity
ControlChange	a CC (fader/knob) message	Matches the controller value
ProgramChange	a program change	Ignored — any PC with the right number fires

The **Vel** field is a small rule, written in plain text:

- leave it blank (or type `any`) — any value fires the cue
- `64` — fires only on exactly 64
- `>0` — fires on anything above 0 (the common choice for note presses)
- `<127` — fires on anything below 127

One quirk worth knowing: per the MIDI spec, a NoteOn with velocity 0 is treated as a NoteOff. If your controller sends "note off" that way, a NoteOff trigger will still catch it.

MIDI Show Control (MSC) — incoming

ATTACCA can act as an MSC slave, taking cues from a lighting or automation console:

1. In **Preferences** → **MIDI**, under **MIDI Show Control (MSC)**, check **Respond to incoming MSC**.
2. Set **MSC Device ID** to the ID your console sends to (0–126). ATTACCA also always answers the all-call ID 127.
3. Click **Save**.

ATTACCA accepts commands addressed to the Sound (General), Lighting (General), Video (General), and All Types formats, and maps them like this:

Incoming MSC command	What ATTACCA does
GO	Fires **GO** (or fires a specific cue if the command names a cue number)
STOP	Pauses (MSC's STOP is a pause, per the spec)
RESUME	Resumes
LOAD	Loads the named cue
ALL_OFF	Panic — fade-stops everything
RESET	Stops everything and returns the playhead to the top
STANDBY+ / SEQUENCE+	Moves the playhead to the next cue
STANDBY− / SEQUENCE−	Moves the playhead to the previous cue

MSC — outgoing

To *send* MSC, use a MIDI cue with its message type set to MSC — you choose the command (GO, STOP, and so on), command format, device ID, and target cue number. See *Show Control Cues* for the full MIDI cue reference. To automatically broadcast MSC whenever cues fire, see the *Broadcast* section at the end of this chapter.

Workspace controls over MIDI

Beyond per-cue triggers, ATTACCA defines 15 workspace-level controls that can be driven by MIDI voice messages. The master switch is in **Preferences → MIDI**, under **Musical MIDI Controls**: check **Use musical MIDI voice messages for workspace controls**.

The mappable controls are:

- **GO**
- **Preview Selected**
- **Audition GO**
- **Audition Preview Selected**
- **Load**
- **Panic Selected**
- **Pause/Resume Selected**
- **Playhead Previous** / **Playhead Next**
- **Playhead Previous Sequence** / **Playhead Next Sequence**
- **Panic All**
- **Pause All**
- **Resume All**
- **Reset All**

None of these is mapped out of the box — a stray note from a keyboard can never fire GO until you deliberately assign it. In this release, Preferences does not yet include a per-control mapping grid; the assignments live in your workspace and travel inside mapping profiles. To set them up, use **Export** in the

Mapping Profiles section to write your current (empty) mappings to an `.atca-midi-profile` file, fill in the controls you want — the file is plain, readable JSON that lists each control by the names above with a message type, note/CC number, and value rule — then **Import** it and click **Save**. An on-screen mapping editor is planned for a future update.

OSC

OSC (Open Sound Control) is ATTACCA's richest remote-control surface — the same protocol QLab speaks, so most OSC controller apps and consoles work out of the box.

Turning the OSC server on

The OSC server is off by default and stays off until you turn it on. A fresh install listens on nothing: no ports are open, no firewall rule exists, and nothing on the network can reach ATTACCA over OSC. (Earlier builds started the server automatically. They no longer do — if you are upgrading and your console suddenly "stopped working", this is why.)

1. Open **Edit** → **Preferences…** (`Ctrl+,`) and select the **OSC** page.
2. Check **Enable OSC Server**.
3. Set a **Passcode** if you want remote *control* — see the next section; without one, remote clients are limited to reading.
4. Click **Save**. The **Server Status** line at the bottom of the page confirms the server is running.

> *[Illustration to be added in a future revision]*

The OSC server requires **ATTACCA Pro**. You can see and change the settings at any tier, but below Pro the server does not start — the status line reads **"Stopped (requires Pro)"**.

Turning it off again during a show is deferred. If you uncheck **Enable OSC Server** while you are in Show Mode or while cues are running, ATTACCA leaves the listener up rather than tearing down something the running show may be using. The status reads **"Running (disabled — takes effect at next launch)"**, and the server is genuinely gone the next time you start the app. Turn it off when the show is idle and it stops immediately.

Server settings

Setting	Default	Notes
Enable OSC Server	**Off**	Master switch — see above
Listen Port (UDP)	53000	Standard OSC over UDP
Enable TCP connections	On	TCP OSC uses SLIP framing (the QLab convention)
TCP Port	53000	Same number as UDP is fine — they're different protocols
Enable plain text connections	On	Accepts plain UTF-8 text like `/go` — no OSC encoding needed
Plain Text Port	53535	This port is **UDP**, not TCP

Setting	Default	Notes
Passcode	empty	See below — required for remote control
Broadcast Destinations	empty	Host/port pairs that receive OSC events when cues fire, stop, or panic
Enable duplicate message filter	Off	With **Filter Interval (s)**, drops repeated messages
UDP Timeout (s)	61	How long an idle UDP client session is remembered

Replies to every command go back as OSC messages addressed `/reply{original-address}` carrying a single JSON string (status, and data for queries). UDP replies are sent to port **53001** on the sender's machine by default.

[Illustration to be added in a future revision]

Passcode and access levels — read this before your console "doesn't work"

ATTACCA grants OSC clients one of two access levels:

- **View** — queries only: reading cue lists, running cues, the playhead position.
- **Control** — everything: firing, stopping, panicking, editing.

Without a passcode, every remote client gets View access only. A touchscreen app can display your cue list, but its GO button will be politely denied. This is deliberate: an open network port that can fire cues in a theater is a hazard, so control always requires a shared secret. The OSC page says as much when the server is enabled with no passcode set: *"No passcode set: remote peers can read status only, and no firewall rule is added. Set a passcode to allow remote control."*

A workspace file cannot grant control on its own. A show file carries a default OSC access level, and opening one is not a way to hand out control of your machine: if a workspace asks for Control and no passcode is set, ATTACCA caps it to **View** and logs a warning saying so. To grant Control you either set a passcode, or raise the level yourself in Preferences — a deliberate act by the person at the keyboard, not something a file can do behind your back.

To enable remote control:

1. In **Preferences** → **OSC**, type a passcode into the **Passcode** field (it's stored only as a salted hash in the workspace file).
2. Click **Save**.
3. In your controller, configure the same passcode. Clients authenticate by sending `/connect` with the passcode as its argument; after a successful `/connect`, that client has Control access. (QLab-compatible apps with a passcode field do this automatically.)

Setting a passcode has a second effect: ATTACCA only creates its inbound Windows Firewall rules (named "ATTACCA OSC …") when a passcode exists. With no passcode, no firewall hole is opened — so on most machines, remote OSC simply won't arrive until you set one. If your network is locked down further, allow ATTACCA through the firewall or open UDP/TCP 53000 (and UDP 53535) on the show network manually.

> **Why it works this way:** an unauthenticated peer on venue Wi-Fi should never be one packet away from firing your cues. View-only-by-default plus passcode-for-control means the worst an intruder can do without the secret is read your cue names.

Addresses ATTACCA answers

The full OSC dictionary is large; these are the workhorses. `{number}` is a cue number as it appears in the Number column.

Address	What it does
`/go`	Fire the standing-by cue (same as pressing **GO**)
`/go/{number}`	Fire a specific cue by number
`/stop`	Fade-stop all running cues
`/hardStop`	Stop everything immediately
`/panic`	Panic (same as `Esc`)
`/pause` / `/resume`	Pause / resume all running cues
`/reset`	Stop everything and return the playhead to the top
`/playhead/{number}`	Move the playhead to a cue (also `/playhead/next`, `/playhead/previous`)
`/select/{number}`	Select a cue (also `/select/next`, `/select/previous`)
`/cue/{number}/start`	Start one cue directly
`/cue/{number}/stop`	Stop one cue (fade-aware; `/hardStop` for immediate)
`/cue/{number}/pause` / `/resume`	Pause / resume one cue
`/new {type}`	Create a cue (Edit Mode; e.g. `/new audio`)
`/cueLists`	Query the full cue list structure (JSON reply)
`/runningCues`	Query what's playing (JSON reply)
`/version`	Query the app version — works even before `/connect`

Cue addresses also accept `selected`, `playhead`, and `active` in place of a number, and wildcards (`/cue/1*/stop`).

Testing OSC with nothing but PowerShell

Because the plain-text port accepts ordinary text, you can test from any Windows machine with no software installed. This example authenticates and fires GO — replace `192.168.1.50` with the ATTACCA machine's IP and `mypasscode` with your OSC passcode, and run it in PowerShell:

```powershell
$osc = New-Object System.Net.Sockets.UdpClient
foreach ($msg in '/connect mypasscode', '/go') {
    $b = [System.Text.Encoding]::UTF8.GetBytes($msg)
    $osc.Send($b, $b.Length, '192.168.1.50', 53535) | Out-Null
    Start-Sleep -Milliseconds 100
}
$osc.Close()
```

The standing-by cue fires. Both messages must be sent from the same PowerShell session (the same socket), because authentication is remembered per connection. If no passcode is set, skip the `/connect` line — but remember that only queries will be accepted, and the firewall rule won't exist, so test from the ATTACCA machine itself using `127.0.0.1`.

Sending OSC from cues

To make ATTACCA the *sender* — firing commands at consoles, media servers, or other ATTACCA machines — use an OSC cue (see *Show Control Cues*; the Network cue that will also cover this is disabled in this release). For automatic fan-out of show events (every cue fire, stop, and panic) to other systems, use **Broadcast Destinations** on the OSC page, or the Broadcast preferences page described at the end of this chapter.

Triggers & arming

The inspector's **Triggers** tab gives every cue up to five automatic ways to fire, listed in the **Trigger Sources** column: **Hotkey**, **MIDI Trigger** (covered above), **Wall Clock**, **Timecode**, and **Audio Marker**. A disarmed cue never fires from any of them.

[Illustration to be added in a future revision]

Hotkey triggers

1. Select the cue and open the **Triggers** tab.
2. Check **Hotkey**.
3. Click the capture box next to it and press the key (or key combination) you want.

The hotkey fires that cue directly — it does not move the playhead, and it works no matter what's selected. Several cues may share one key; they all fire together.

Some keys are reserved and will not be accepted: `Esc` (always Panic), the `Up` / `Down` arrows, and bare `Ctrl` / `Alt` / `Shift` presses.

Wall-clock triggers

A wall-clock trigger fires a cue once per day when the clock reaches a set time — house music at 7:00 PM, walk-in lights at half-hour.

1. Check **Wall Clock** on the Triggers tab.

2. Set the time with the **Hour : Min : Sec** boxes and the AM/PM dropdown. The line below ("Times shown in: …") tells you which timezone applies — that's the system clock unless you've set a manual override in **Preferences → General** under "Clock".

In plain terms: when the workspace clock ticks past the set time, the cue fires within a few seconds. It fires once, then not again until tomorrow. If ATTACCA wasn't running — or triggers weren't armed (see below) — at that moment, the cue simply doesn't fire that day; there is no catch-up firing later. One known edge: a time inside the hour skipped by a spring-forward DST change won't fire on that particular day.

Timecode triggers (not available in this version)

Be direct about this one: **ATTACCA 1.0 does not chase incoming timecode.** The **Timecode** trigger on the Triggers tab and the **Input / Chase** section of **Preferences → Timecode** (Enable timecode input, Sync Source, MIDI Input Device, Audio Input Channel, SMPTE Format, On Start, On Stop, Freewheel Time) are present and can be viewed and saved, but no incoming MTC or LTC is decoded in this version, so a cue with a Timecode trigger never fires from timecode and the **Incoming** readout in the **Timecode Status** section stays empty. Workspaces that contain timecode triggers load normally — the triggers simply do nothing until a future update connects the timecode input.

What works today is the other direction: ATTACCA **generates** MTC and LTC for other devices to chase, from a Timecode cue — see *Timecode cues* in *Show Control Cues*.

[Illustration to be added in a future revision]

Audio-marker triggers

Any cue can fire when another cue's audio playback crosses a named marker — check **Audio Marker** and pick the **Source** audio cue and the **Marker**. Markers are created on the audio cue's waveform; see *Audio in Depth* for the marker workflow. Seeking past a marker does not fire it — only natural playback does.

Behaviors: what a second press does

The right-hand **Behaviors** column of the Triggers tab includes **Second Trigger** — what happens if the cue's trigger fires again while the cue is already running: **DoNothing**, **Panic**, **Stop**, **HardStop**, or **HardStopAndRestart**, plus a **Second trigger on release** checkbox for note-off/key-up behavior. The same column holds Fade & Stop Others and Duck/Boost, covered in *Audio in Depth*.

Arming — when triggers are live

This is the part that surprises people, so here it is plainly:

- **When a workspace loads, wall-clock triggers are disarmed.** Nothing fires on a schedule just because you opened the file.
- **They arm when you enter Show Mode.** The Show/Edit toggle is the arming control — there is no separate "Arm Triggers" button.
- **Editing any trigger in the inspector also arms them**, so a trigger you just created can be tested immediately without flipping modes.

- **Returning to Edit Mode does not disarm.** Once armed, triggers stay armed until the workspace is closed or reloaded — a mid-show trip to Edit Mode to fix a level won't kill your timecode chase.

Hotkey triggers are always registered (they need a human keypress anyway), and MIDI and audio-marker triggers are governed by each cue's own armed state.

> **Why it works this way:** a workspace opened for editing shouldn't fire cues because a clock struck. Tying arming to Show Mode means the moment you declare "this is a performance," the schedule goes live — and not one moment before.

Web Remote

The Web Remote is a browser-based monitor and remote control served straight from ATTACCA — nothing to install on the phone or tablet.

What it serves

- **View Mode** at `/` — a read-only monitor: show name, the current playhead cue, active cues with progress bars, and the last 10 fired cues. No control buttons at all.
- **Control Mode** at `/control` — the full cue list plus **GO**, **STOP**, **PAUSE**/**RESUME**, and **PANIC** buttons (PANIC asks for confirmation), and tap-a-row to move the playhead.

What it deliberately cannot do in v1: no editing, no creating or deleting cues, no level changes, no saving. Anything but the transport actions above is rejected, and commands are rate-limited to 10 per second per device.

[Illustration to be added in a future revision]

[Illustration to be added in a future revision]

Enabling it and reaching it from a phone

The settings live in **Preferences** → **Network** (the page is headed **Web Remote Monitor**):

1. Check **Enable Web Remote**. On its own, this binds to `127.0.0.1` — the remote is reachable **only from the ATTACCA machine itself**, and the toolbar URL shows "(local only)".
2. Check **Allow remote (network) access**. This is the switch that makes the server listen on the network; it also automatically adds a Windows Firewall rule named "ATTACCA Web Remote" (and removes it when the server stops). If you have multiple network adapters, pick the show network in the **Network Interface** dropdown.
3. Set the **Port** if 8080 is taken (1024–65535).
4. Set a Web Remote password in the **Password** / **Confirm** boxes. This password is **independent of the Show Mode password**. Control Mode always requires it — ATTACCA will refuse to save with **Enable Control Mode** on and no password set. Optionally check **Require Password for View Mode** to protect the monitor page too.

5. If you want GO from the phone, check **Enable Control Mode** (it's off by default).
6. Click **Save**.
7. The main-window toolbar now shows the live URL, e.g. `http://192.168.1.50:8080`. On a phone **on the same Wi-Fi network**, type that URL into the browser; add `/control` for the control page and enter the Web Remote password when prompted.

[Illustration to be added in a future revision]

[Illustration to be added in a future revision]

A few practical notes:

- There is no QR code — you type the URL the toolbar shows. It's short on purpose.
- A control login lasts **8 hours**, so one sign-in covers the longest tech day. Changing the password immediately signs out every connected device.
- The page updates live (cue fires, progress, playhead moves) over a WebSocket — no refreshing needed.

Lua scripting & Script cues

A Script cue runs a small Lua program when it fires — for the odd bit of conditional logic that plain cues can't express. Script cues are Pro-tier.

What scripts can do

Each script runs in a fresh, sandboxed Lua environment with this API:

Call	What it does
`log("text")`	Writes to the cue's Output console (`print` goes there too)
`wait(seconds)`	Pauses the script (the show keeps running; stopping the cue cancels the wait)
`cue("10")`	Finds a cue by number — returns a cue object, or `nil` if not found
`cue_id("…")`	Finds a cue by its unique ID
`workspace.go()`	Fires GO
`workspace.stop_all()`	Stops all running cues
`workspace.panic()`	Panic

A cue object found with `cue()` supports (colon syntax): `:go()`, `:stop()`, `:pause()`, `:resume()`, and getters/setters `get_name` / `set_name`, `get_number` (read-only), `get_armed` / `set_armed`, `get_color` / `set_color`, `get_type` (read-only), `get_duration` / `set_duration`, `get_prewait` / `set_prewait`, `get_postwait` / `set_postwait`, `get_notes` / `set_notes`, `get_level` / `set_level` (an audio cue's main level, in dB), and `get_rate` / `set_rate` (audio playback rate).

Sandbox limits

Scripts are for show logic, not system administration. There is no file access, no OS access, no loading of external modules — Lua's `io`, `os`, `require`, and friends are removed; the standard `string`, `math`, `table`, `coroutine`, and `utf8` libraries remain. Each Script cue has a **Timeout** field (default **5 s**, settable 0.1–600); a script that runs past its budget is stopped with "Script timed out". A runaway script can therefore never hang your show.

Writing and testing a script

Scripts are written on the Script cue's **Script** inspector tab: a code editor, a **Timeout** field, an error area, and an **Output** console, with three buttons — **Run** executes the script right now (without firing the cue through the engine), **Validate** checks the syntax, and **Clear Output** empties the console.

Try this — it finds cue 1, fires it, and logs what it did:

```
local c = cue("1")
if c then
    c:go()
    log("Fired cue " .. c:get_number())
end
```

Click **Run** and watch the Output console.

[Illustration to be added in a future revision]

Scripts or cues?

Reach for a cue first. Cues are visible in the list, undoable, panic-safe, and legible to whoever operates the show after you. A Group cue with waits covers most "do several things" needs. Use a script when you genuinely need a decision — "if the walk-in music is still playing, fade it, otherwise go straight to the preshow announcement" — or a calculated value that no fixed cue property can express.

Collaboration — sharing your workspace with observers

Collaboration in this release is deliberately modest: **one Primary, read-only Observers**. It's a "follow along" display for a director, designer, or calling SM on another machine — not a second console.

Open it from **View → Collaboration...**.

[Illustration to be added in a future revision]

Sharing (the Primary)

1. Set your **Display Name:** (how you'll appear to observers).
2. Under **Share This Workspace**, keep the default **Port:** (53600) unless it clashes, and set a **Passcode:** — it is **required**: sharing refuses to start without one ("Set a passcode to share — observers must enter it to

connect"), and it is stored only as a salted hash. It is independent of the Show Mode and Web Remote passwords.
3. Click **Start Sharing**.

The window switches to "Sharing as Primary" and shows a **Connected Observers** table (**Name** / **IP** / **Connected**). Up to **20 observers** can join. **Stop Sharing** disconnects everyone.

Joining (an Observer)

1. On the other machine, open **View → Collaboration…** and set a **Display Name:**.
2. Under **Connect to Workspace**, shared workspaces on the local network appear automatically in the list (**Workspace** / **Host** / **Observers** columns). Select one, enter the **Passcode:** the primary set, and click **Connect to Selected**. If discovery can't see the primary (see *Common issues*), type its address into the **Manual:** box as `host:port` and click **Connect**.
3. The window switches to "Connected as Observer" with a **READ-ONLY** badge; **Disconnect** leaves the session.

> *[Illustration to be added in a future revision]*

What observers actually see — the honest version

On connect, the observer receives a one-time snapshot of the whole workspace (with all password hashes stripped out) and loads it. From then on, two things stay live: **the primary's playhead position** and **individual cue property edits** (rename a cue, change a level — the observer sees it).

What does **not** mirror: cue firing, stopping, and panic are not replayed on the observer's machine, and cues added or deleted on the primary do not appear or vanish on the observer — the structure is frozen at connect time (reconnect to get a fresh snapshot). Observers cannot push any change back. Plan around this: it's a synchronized script to read along with, not a mirror of the running show.

Broadcast — announcing show events to other systems

Preferences → Broadcast makes ATTACCA *emit* show-control messages automatically as the show runs, with no cues to program:

- **MSC (MIDI Show Control) Output** — check **Enable MSC broadcast** and every cue fire sends an MSC GO, Panic/Stop All sends ALL_OFF, and Reset All sends RESET, out your enabled MIDI output. Set the **Device ID (0-127)** (default 127 = all-call), **MIDI Patch**, and **Command Format** (All Types, Sound (General), Lighting (General), Machinery, Video, Projection, Process Control, or Pyro) to match the listening console.
- **OSC Broadcast** — check **Enable OSC broadcast** and set **Destination Host** and **Destination Port** (default 53001). ATTACCA sends OSC events for GO, cue starts and stops, panic, stop all, and pause/resume all. External systems can also subscribe themselves at runtime with the OSC `/listen` command, and the event format can be customized via `/eventFormat`.

> *[Illustration to be added in a future revision]*

Common issues

- **A phone can't open the Web Remote, but it works on the ATTACCA machine.** The toolbar URL says "(local only)": **Allow remote (network) access** is unchecked in Preferences → Network. Check it, Save, and use the `http://<ip>:8080` URL the toolbar then shows.
- **Web Remote or OSC works on some networks but not the venue Wi-Fi.** Many venue and guest networks enable *AP isolation* (also called client isolation), which blocks devices from talking to each other even on the same SSID. Ask for the production network, or use a dedicated show router.
- **Everything breaks when the ATTACCA machine's VPN is on.** A VPN reroutes traffic away from the local network and usually blocks multicast/broadcast, killing collaboration discovery, sACN multicast, and often the Web Remote and OSC. Disconnect the VPN (or exclude the show subnet) on show machines.
- **Nothing at all reaches ATTACCA over OSC.** The server is off by default — check **Enable OSC Server** in Preferences → OSC and Save. If the status line reads "Stopped (requires Pro)", the OSC server needs a Pro license.
- **I unchecked Enable OSC Server but it's still listening.** Turning OSC off while a show is running is deferred to the next launch on purpose (the status says "Running (disabled — takes effect at next launch)"). Restart ATTACCA, or turn it off while nothing is running.
- **An OSC app connects but every GO is "denied".** No passcode is set, so remote clients get View-only access. Set a **Passcode** in Preferences → OSC and enter it in the app (it must send `/connect` with the passcode — QLab-compatible apps do).
- **OSC packets never arrive from another machine.** With no passcode set, ATTACCA doesn't create its firewall rules. Set a passcode, or allow the app through Windows Firewall manually. Also note the plain-text port 53535 is UDP — sending TCP text to it does nothing.
- **`/control` says to set a password.** The Web Remote password is mandatory for Control Mode — set it in Preferences → Network. It is not the Show Mode password.
- **A MIDI trigger won't fire.** Check, in order: the device's checkbox is enabled in Preferences → MIDI; **Enable MIDI input (receive)** is on; the cue is armed; the trigger's **Ch** matches (0 follows the workspace **MIDI Channel**); and the **Vel** rule isn't excluding your controller's values.
- **The Learn button is greyed out.** No MIDI input device is enabled yet.
- **A wall-clock trigger didn't fire.** Triggers are disarmed when a workspace loads and arm when you enter Show Mode (or edit a trigger). If the set time passed while disarmed, that day's firing is skipped — there's no retroactive fire.
- **Esc won't record as a hotkey.** `Esc`, the `Up` / `Down` arrows, and bare modifier keys are reserved and can't be assigned as cue hotkeys — `Esc` is always Panic.
- **Observers don't see cues fire.** By design in v1 — observers get a snapshot plus live playhead and property edits only. Cue fires, stops, and added/deleted cues do not mirror.
- **No shared workspaces appear in the Collaboration list.** UDP discovery is blocked: different subnet, AP isolation, VPN, or firewall. Use the **Manual:** box with the primary's IP and port (default 53600) instead.

Running the Show

Everything in this chapter happens after the design work is done: locking the workspace for performance, stopping everything in a hurry, keeping the stage manager's clock running, and making sure the machine itself doesn't surprise you at 8:02 PM.

Show Mode

ATTACCA has two mutually exclusive modes. **Edit Mode** is where you build; **Show Mode** is where you perform. Switching to Show Mode locks the workspace against accidental changes while leaving every playback control live.

Entering Show Mode

Any of these works:

1. Click the **Show** toggle at the bottom-left of the main window (next to **Edit**, in the status bar).
2. Press `Ctrl+Shift+]` (Enter Show Mode). `Ctrl+E` toggles between the two modes, and `Ctrl+Shift+[` returns to Edit Mode.
3. Choose **View → Edit Mode** to toggle the current mode.

You can tell at a glance which mode you're in:

- The title bar reads `[Show]` instead of `[Edit]`.
- A red **SHOW MODE** badge appears at the bottom-right of the window, and a translucent red wash covers the toolbar.

[Illustration to be added in a future revision]

What Show Mode locks

- The entire **Edit** menu is disabled — no Undo/Redo, Cut/Copy/Paste, Duplicate, Delete, or Preferences.
- Cues cannot be created, deleted, moved, renumbered, or edited. The inspector is hidden.
- **File → New / Open… / Save / Save As…** are unavailable.
- **DMX Status…** and **Light Dashboard…** cannot be opened (they are Edit Mode tools).
- The Relink Media dialog will not pop up mid-show — missing-media prompts are suppressed until you return to Edit Mode.

What is *not* locked: everything you need to run the show. **GO**, Stop, Pause/Resume, Panic, playhead navigation, the Show Timer, the Web Remote, and all incoming control (OSC, MIDI, hotkeys) keep working. Show Mode is the performance mode, not a freeze.

Entering Show Mode also arms your wall-clock triggers so a scheduled cue can actually fire — see *Control & Integration* for the full arming model.

The exit password

For situations where the operator should not be able to wander back into Edit Mode — volunteer boards, long-running installations — you can require a password to leave Show Mode:

1. In Edit Mode, open **Edit → Preferences…** (`Ctrl+,`) and go to the **General** page.
2. Under **Show Mode Security**, check **Require password to exit Show Mode**.
3. Enter the password twice (the second box confirms it) and click **Save**.

With the password set, any attempt to leave Show Mode opens a dialog titled **Enter Password to Return to Edit Mode**. A wrong entry shows *Incorrect password* and the workspace stays locked; **Cancel** stays in Show Mode.

> *[Illustration to be added in a future revision]*

The password is stored in the workspace file as a salted cryptographic hash — it is not recoverable from the file. If it is ever forgotten, there is a deliberate recovery mechanism documented in *Troubleshooting, Known Limitations & FAQ*.

The panic system

Panic is ATTACCA's "everything off, gracefully" button, and it deserves its own section because it is the one control you will reach for under pressure.

Single panic: `Esc`

Press `Esc` — or click the red **Panic** button in the toolbar — and everything running fades out together over the **panic fade time**:

- All running audio, video, and camera cues fade-stop. Video output windows fade to black and close; camera hardware is released (the warm pool survives, so a camera cue can be re-fired instantly).
- On any display with **Standby Black** enabled (Preferences → Displays), the picture fades to **black rather than back to the desktop** — the black backer stays up. That is the point of the option: a panic in front of an audience should not reveal your wallpaper.
- Lighting fades to black over the same time, and running lighting effects fade out with it.

The panic fade time is set in **Edit → Preferences… → General**, section **Panic (ESC)**, field **Fade Duration**. The presets are **Instant (0ms)**, **250ms**, **500ms**, **1000ms**, **2000ms**, and **Custom…** (0–10000 ms). The default is **500ms** — fast enough to feel immediate, slow enough not to slam the room into silence.

> *[Illustration to be added in a future revision]*

Holding `Esc` down fires panic once, not repeatedly — a stuck or leaned-on key won't escalate on its own.

Double panic: hard stop and blackout

A **second panic within 1 second** of the first escalates: every running cue stops *immediately* (no fade), lighting effects halt, and DMX output is blacked out — every channel on every universe driven to zero, held there until the next cue writes something. Tap `Esc` twice and the show is dark and silent, full stop.

The 1-second window counts panics from **any source**. `Esc` followed by an OSC `/panic`, or a MIDI panic mapping followed by `Esc` — any two within a second escalate to the hard stop. Remote controllers and the operator's keyboard are equal citizens.

> **Why it works this way:** a one-stage panic forces a bad choice at design time — either it's abrupt (ugly for the audience on a false alarm) or it's gentle (too slow in a genuine emergency). Two stages let the first press be graceful and reversible-feeling, while the instinctive second press — the one you make when something is actually wrong — is instant and absolute.

Where panic works — and the one place it doesn't

- **From the cue list / main window:** `Esc` always panics, even when no particular control has keyboard focus. (The same low-level hook keeps `Space` = GO and bare `S` = Stop Selected working; be aware that a stray `S` press with the main window focused will stop the selected cue.)
- **Standby Black backers and test-pattern cards are completely inert.** Clicking one — or, on a touch monitor, touching it — does nothing and never takes focus off the cue list: GO, `Space`, and `Esc` keep working. `Ctrl+Shift+F12` closes them from anywhere regardless of what has focus.
- **The video output window does not take keyboard input.** It is deliberately non-focusable so it can never steal keystrokes from the operator — which also means pressing `Esc` "at" the video display does nothing. Keep focus on the main window and panic from there. (If a video window ever ends up covering your main window — see *Troubleshooting, Known Limitations & FAQ* — clicking into it and then pressing `Esc` does panic.)
- **Remote panic:** OSC `/panic` and any MIDI panic mapping trigger exactly the same escalation logic.
- **The caveat that matters:** `Esc` does **not** panic while the text cursor is in a text field — a cue name, a notes box, the search bar. `Esc` is reserved for cancelling the edit there. Click out of the field or press `Enter` / `Tab` first, then `Esc` works. During a performance, keep focus on the cue grid or empty window space — those are always safe.

The global kill hotkey: `Ctrl+Shift+F12`

`Ctrl+Shift+F12` is registered with Windows itself, so it works **system-wide** — even when ATTACCA isn't the focused application, and even if ATTACCA's interface has stopped responding. It does one thing: **immediately closes every output window** — video windows, Standby Black backers, and any test-pattern card, so the desktop comes back on every display. It does not stop audio or lighting — it is the "get that image off the screen *now*" control for when a projector is showing something it shouldn't and you can't reach the app. Follow it with a normal panic once you're back in ATTACCA.

This hotkey is fixed (it isn't in the Keyboard Shortcuts list) and cannot be remapped.

Alignment before the house opens

The **Test Pattern** checkbox on Preferences → Displays throws an alignment card onto a display immediately — grid, crosshair, corner markers, edge border, and a readout of the display's name and pixel size. Use it to focus a projector or align an LED region, then **turn it off before the house opens**: a video cue covers the card while it plays, but the card comes back between cues. While any card is up, the Displays page carries an amber **TEST PATTERN ACTIVE** banner as a reminder, and the cards clear on their own at the next app restart.

[Illustration to be added in a future revision]

[Illustration to be added in a future revision]

The Show Timer

The Show Timer is a large-format clock window built to be read from across a booth — or mirrored to a monitor backstage.

Opening it

Choose **View** → **Show Timer** or press `Ctrl+T` (pressing it again closes the timer).

[Illustration to be added in a future revision]

Top to bottom, the window shows:

- **Message bar** — an operator message ("PLACES", "HOLD — LATE SEATING") shown across the top when one is active.
- **Current time** — the wall clock, in 12- or 24-hour format with optional seconds.
- **SHOW ELAPSED** — a count-up timer for total running time.
- **COUNTDOWN** — a count-down timer with an optional label; it reads `–` when idle and flashes when it hits zero.
- **Next cue** — the number and name of the cue standing by at the playhead.
- **Controls strip** — the manual control panel (can be hidden or detached).

Every section can be shown or hidden, recolored, and resized independently (see the settings dialog below). The timer's math is wall-clock based, so it stays accurate even if the machine sleeps and wakes. All appearance settings are saved in the workspace, so your venue setup travels with the show file.

Right-click is the way in

The Show Timer has no menus, no gear icon, and no buttons for its own options. **Everything is on the right-click context menu of the timer window** — this is the only way in. The items, in exact order:

1. **Always On Top** (checkable — on by default)
2. **Fullscreen** (checkable)
3. **Show Controls** (checkable — shows/hides the controls strip at the bottom)
4. **Detach Controls** (checkable — moves the controls into their own floating window)

5. **Reset Show Timer**
6. **Stop Countdown**
7. **Timer Settings…** — opens the settings dialog
8. **Close**

[Illustration to be added in a future revision]

Timer Settings

Right-click the timer and choose **Timer Settings…**. The **Show Timer Settings** dialog applies every change live as you make it — there is no OK/Cancel, just a **Close** button when you're done.

[Illustration to be added in a future revision]

Display

- **Always on top** (on by default)
- **Fullscreen**
- **Show controls** (on by default)
- **Detach controls into separate window**
- **Show seconds** (on by default)
- **Time format** — **24-hour (HH:MM:SS)** or **12-hour (h:MM:SS AM/PM)**. This is where you switch the clock to 12-hour time; the default is 24-hour.
- **Display windows** — 1 to 4 (see mirror windows below).

Visible sections — check or uncheck **Current time**, **Elapsed timer**, **Countdown**, and **Next cue info** independently. Want a pure countdown display for the deck? Turn everything else off.

Appearance — the remaining groups style each section:

- **Background** — color and an opacity slider (0–100%).
- **Current Time** — color and font size (default 72).
- **Elapsed Timer** — color, font size, and label color.
- **Countdown** — color, font size, label color and label font size, plus **Flash on zero** (on by default) and the flash color.
- **Next Cue Info** — color and font size.
- **Message Display** — color, font size, and background color.

Font-size boxes offer presets from 10 to 200 and also accept a typed value.

Fullscreen — and how to get back out

Right-click the timer and check **Fullscreen**. The timer fills the entire monitor, covering the taskbar, and its title bar disappears (move the mouse to the very top edge of the screen and the title bar slides back into view).

To exit fullscreen, right-click the timer and click Fullscreen again to uncheck it (or uncheck **Fullscreen** in Timer Settings). There is **no `Esc` exit** — the Show Timer deliberately ignores the keyboard so that `Esc` stays what it always is during a show: panic. The window returns to exactly the size and position it had before.

[Illustration to be added in a future revision]

The controls

The controls strip (bottom of the timer window) has three labeled rows:

- **Elapsed:** — **Start**, **Stop**, **Reset** for the count-up show timer.
- **Countdown:** — a duration box (accepts `MM:SS`, `HH:MM:SS`, or plain seconds), a **Label:** field (e.g. "INTERMISSION"), then **Start**, **Pause** (which becomes **Resume**), and **Stop**.
- **Message:** — a text box with **Show** and **Clear**.

Check **Detach Controls** on the right-click menu to pop the same panel into its own small floating window titled **Timer Controls** — useful when the timer itself is fullscreen on another monitor. The detached window has a **Reattach Controls** button at bottom-right; the two stay in sync because they are the same controls.

[Illustration to be added in a future revision]

A worked intermission: type `15:00` in the countdown box, label it `INTERMISSION`, click **Start**. The countdown runs on the timer (and every mirror). At zero, the countdown flashes — house management can see the act break is over from anywhere the timer is displayed.

Mirror windows

Set **Display windows** in Timer Settings to 2, 3, or 4 and additional copies open, titled **Show Timer (2)** through **Show Timer (4)**. Drag them to other monitors — booth, deck, green room. Mirrors are display-only: they show exactly what the primary timer shows and never have a controls strip. All windows share one set of appearance settings.

[Illustration to be added in a future revision]

Driving the timer from the cue list

Everything the manual controls do, a **Timer cue** can do from the cue list — start the show timer on the first GO, launch the intermission countdown from the act-one blackout cue, or post an operator message automatically. See *Control & Structure Cues* for the Timer cue's actions.

Pre-show checklist

A grounded routine for the half-hour call. Adapt it to your venue.

1. **Open the workspace well before the house opens.** Opening early gives autosave-recovery and the Relink Media dialog a chance to appear and be dealt with — both are suppressed or disruptive once you're mid-

show. If the Relink dialog appears, resolve it now (see *Workspaces & Files*-related coverage of relinking in the media chapters).

2. **Verify the audio output device. View → Audio Output Device** — confirm the checkmark is on the venue's interface, not a laptop speaker or a Bluetooth device that paired itself. Fire a known cue and confirm level at the console.
3. **Fire a test cue to every display.** One short video or image cue per display number confirms the projector routing and that the display mapping (**Preferences → Displays**) matches how the venue's cables actually landed today.
4. **Check DMX output.** Open **View → DMX Status…** (Edit Mode) and confirm channels move when you fire a light cue. Close it before entering Show Mode — it isn't available there.
5. **Enter Show Mode and verify triggers.** Entering Show Mode arms wall-clock triggers; confirm any scheduled cue shows the behavior you expect, and confirm cues that must respond to MIDI/hotkeys are Armed.
6. **Silence Windows.** Turn on Focus Assist / Do Not Disturb, and set Power & sleep so the machine never sleeps or turns off the display. A toast notification over your video output — or a machine that dozes off during a 20-minute underscore — is a preventable disaster.
7. **Close other audio applications**, especially anything that may take exclusive control of the audio device (some DAWs, conferencing apps, browser tabs with hardware acceleration).
8. **Confirm your license state.** If the show uses licensed cue types (video, lighting, MIDI…), confirm **View → License…** shows the expected tier now — a locked cue at places is the wrong time to discover a license issue.
9. **Do not hot-unplug displays once cues are running.** ATTACCA re-resolves monitors on the next GO, but a live disconnect is not detected mid-cue (see *Troubleshooting, Known Limitations & FAQ*). Cables seated, adapters taped.

Running offline

ATTACCA is fully offline-capable. There is no license "phone home" — your license is validated locally from a file on your machine — and updates are optional and manual: the startup update check (Preferences → General → **Check for updates on startup**) can simply be turned off, or the machine can just not have internet. Nothing in the app stops working, times out, or nags without a connection.

What *does* need a network is exactly what you'd expect — the features whose entire job is talking to other devices: OSC in/out, Art-Net and sACN lighting output, NDI, the Web Remote, Collaboration, and MIDI over network transports. These need a network (usually just your closed show LAN — still no internet), plus two genuinely internet-dependent conveniences: the update check and the Open Fixture Library download in the Fixture Library window.

An air-gapped show machine on an isolated lighting network is a fully supported — arguably ideal — configuration.

Common issues

- **`Esc` didn't panic** — your cursor was in a text field (name, notes, search). Click the cue grid or press `Enter` / `Tab` to leave the field, then `Esc`. During a show, keep focus on the cue grid.
- **Pressed `Esc` at the video screen and nothing happened** — the video output window doesn't take keyboard input by design. Panic from the main window.
- **A cue stopped by itself** — check for a stray bare `S` press: `S` is Stop Selected and works even when nothing has focus.
- **The Show Timer is stuck fullscreen** — right-click it and click **Fullscreen** to uncheck it. There is no `Esc` exit.
- **Can't find the Show Timer's settings** — there is no gear button; right-click the timer window and choose **Timer Settings...**.
- **Can't leave Show Mode** — a Show Mode exit password is set. If it's been lost, see *Troubleshooting, Known Limitations & FAQ*.
- **A scheduled cue didn't fire in rehearsal** — time-based triggers arm when you enter Show Mode (or when you edit a trigger). If you loaded the workspace and stayed in Edit Mode without touching a trigger, they're disarmed by design.

Settings Reference

This chapter is the complete reference for every setting in ATTACCA. Preferences is where almost all of them live: open it with **Edit** → **Preferences...** or `Ctrl+,` (the shortcut is remappable). The Preferences window has a sidebar with eleven pages, covered below in sidebar order. Two groups of settings live outside Preferences — the Show Timer's own settings dialog and the License window — and they are covered at the end of this chapter.

Licensed pages. The **Video** and **Network** pages configure Standard features; **OSC**, **MIDI**, **Timecode**, **Lighting**, and **Broadcast** configure Pro features. Reaching one of these from its own **View** menu item below your tier shows the upgrade dialog instead of opening it. The pages themselves remain visible in the Preferences sidebar at every tier — you can read and change the settings; what your license governs is whether the feature actually runs. **General**, **Audio**, **Displays**, and **Keyboard Shortcuts** are unlicensed throughout.

How the Preferences window behaves

- **Save** is disabled until you change something. Clicking it validates and saves every page, and the window **stays open** so you can keep working.
- **Cancel** reverts all pending changes and closes the window.
- Closing the window with unsaved changes asks **"Save changes to preferences?"** (Yes / No / Cancel).
- `Esc` closes the window (except while the Keyboard Shortcuts page is waiting for you to press a new key — there it cancels the capture instead).
- Most settings are saved **into the workspace file** (`.atca`), so they travel with your show. The exceptions are per-machine: the update-check toggle, your keyboard shortcuts, and your license.

> **Why it works this way:** show settings like OSC ports, lighting output, and display mappings belong to the *show*, not the computer. Saving them in the workspace means the show behaves identically when you move the `.atca` file to a different machine.

General

Header: **General Settings**.

[Illustration to be added in a future revision]

Setting	What it does	Options / range	Default
App Name / **Version** / **.NET Version**	Read-only information about this build.	—	—

Setting	What it does	Options / range	Default
Check for updates on startup	Checks GitHub for new releases when the application starts, at most once per 24 hours (see *Reference: Shortcuts, Files & Formats* for the full update flow). Stored per-machine.	on / off	On
Fade Duration (section **Panic (ESC)**)	How long the panic fade lasts when you press `Esc` — every running cue fades out over this time.	**Instant (0ms)**, **250ms**, **500ms**, **1000ms**, **2000ms**, **Custom…**	**500ms**
Custom (ms)	Custom panic fade length in milliseconds. Only visible when **Custom…** is selected.	0–10000	5000
Interval (section **Autosave**)	How often ATTACCA writes a companion `.autosave` file to protect your last explicit save. **Disabled** turns autosave off.	**30 seconds**, **1 minute**, **2 minutes**, **5 minutes**, **10 minutes**, **Disabled**	**1 minute**
Time Source (section **Clock**)	Which timezone wall-clock cue triggers use.	**System Clock (Local)**, **Manual Timezone Override**	**System Clock (Local)**
Timezone	The override timezone. Only visible when **Manual Timezone Override** is selected.	all Windows timezones	none
Active Timezone / **Current Time**	Read-only: the timezone actually in effect and a live clock (12/24-hour format follows the Show Timer's time-format setting).	—	—
Require password to exit Show Mode (section **Show Mode Security**)	When enabled, a password is required to return to Edit Mode from Show Mode. The password is stored in the workspace as a salted hash — it cannot be read back out of the file.	on / off	Off
Password / **Confirm**	Set or change the Show Mode password (visible when the checkbox is on). If a password is already set, the helper reads "Password is set. Enter a new password below to change it."	—	empty
Log Directory (section **Logging**)	Read-only: where ATTACCA writes its log files.	—	`%APPDATA%\ATTACCA\logs`

There is no theme setting — ATTACCA is dark-theme only.

Audio

Header: **Audio Settings**.

[Illustration to be added in a future revision]

Setting	What it does	Options / range	Default
Audio Output Device	Every detected output device, with its type and channel count; the Windows default carries a **Default** badge and the active device a checkmark. **Selecting a device switches to it immediately** — this is the one Preferences change that does not wait for **Save** and is not undone by **Cancel**.	detected devices	current device
Current Device (section **Status**)	Read-only: "Current: {device}" or "No audio device active".	—	—
Latency Mode (section **Audio Latency**)	Trade-off between audio responsiveness and stability (buffer size). Lower = snappier GO, higher = more resilient on busy machines. **Changes take effect on next app restart.**	**Low (50ms)**, **Medium (100ms)**, **High (200ms)**	**Medium (100ms)**
Refresh Devices	Re-scans for output devices (use after plugging in an interface).	—	—

OSC (Pro)

Header: **OSC Settings**. Controls the built-in OSC server that lets consoles, touch panels, and other software query and control ATTACCA (see *Show control protocols* material elsewhere in this manual for the command dictionary).

[Illustration to be added in a future revision]

Setting	What it does	Options / range	Default
Enable OSC Server	Master switch for the OSC server. **Off by default** — a fresh install has no OSC listener at all. Unchecking it while a show is running defers the shutdown to the next launch (the Status line says so).	on / off	**Off**
Listen Port (UDP)	UDP port ATTACCA listens on for OSC.	1024–65535	53000
Reply Port (UDP)	UDP port replies are sent back to.	1024–65535	53001
Enable TCP connections	Accept OSC over TCP (SLIP-framed) as well as UDP.	on / off	On

Setting	What it does	Options / range	Default
TCP Port	TCP port for OSC. It defaults to the same number as the UDP port — that is fine, since they are different protocols.	1024–65535	53000
Enable plain text connections	Accept plain-text (non-SLIP) OSC-style messages — handy for simple controllers and test tools.	on / off	On
Plain Text Port	Port for plain-text connections.	1024–65535	53535
Passcode (section **Security**)	Passcode remote clients must present (via `/connect`) before they can fire or edit cues. Stored as a salted hash in the workspace file. Leave blank to keep the existing passcode; type a new value to change it. **Without a passcode, remote clients can query but not control** — read-only is the secure default, and a workspace file cannot raise its own access level to Control without one.	up to 64 characters	empty
Broadcast Destinations	Send OSC events to external devices when cues fire, stop, or panic. Add a **Host** (e.g. `127.0.0.1`) and **Port** (1024–65535), then **Add**; leave the list empty to disable broadcasting.	host + port list	empty
Enable duplicate message filter (section **Advanced**)	Ignores identical OSC messages that arrive in rapid succession (some controllers double-send).	on / off	Off
Filter Interval (s)	The duplicate-detection window, in seconds.	≥ 0.0	0.1
UDP Timeout (s)	How long an idle UDP client session is remembered.	10–600	61
Server Status / Connected Clients (section **Status**)	Read-only live status of the server and its clients: `Stopped`, `Stopped (disabled)`, `Stopped (requires Pro)`, `Starting…`, `Running`, or `Running (disabled — takes effect at next launch)`.	—	—

MIDI (Pro)

Header: **MIDI Settings**.

[Illustration to be added in a future revision]

Setting	What it does	Options / range	Default
Enable MIDI	Master switch for the MIDI engine.	on / off	On

Setting	What it does	Options / range	Default
Mapping Profiles	Save your whole MIDI setup under a name and recall it later — buttons **Load**, **Delete**, **Save Profile**, **Export**, **Import**. Profiles are stored per-machine in `%APPDATA%\ATTACCA\midi-profiles\`.	saved profiles	none
Enable MIDI input (receive)	Accept incoming MIDI (triggers, MSC, workspace controls).	on / off	On
Enable MIDI output (send)	Allow ATTACCA to send MIDI (MIDI cues, MSC broadcast).	on / off	On
MIDI Devices — Input Devices / Output Devices	One checkbox per detected device. **All devices start unchecked — you must opt in each device you want ATTACCA to use.** Shows "(No MIDI input devices detected)" / "(No MIDI output devices detected)" when nothing is found.	detected devices	all off
MIDI Channel (section **Workspace Channel**)	Filters incoming workspace-control MIDI: 0 = Any channel, 1–16 = specific channel.	0–16	0
Use musical MIDI voice messages for workspace controls (section **Musical MIDI Controls**)	Lets note/CC messages drive transport actions (GO, panic, playhead moves…). The individual mappings all start unassigned.	on / off	Off
Respond to incoming MSC (section **MIDI Show Control (MSC)**)	Act on incoming MIDI Show Control commands (GO, STOP, ALL_OFF…).	on / off	Off
MSC Device ID	The device ID ATTACCA answers to: 0–126 (it also responds to all-call 127).	0–126	0
Middle C (section **Note Display**)	How MIDI note 60 is labeled across the app — match your other software.	**C4 (Traktor, Resolume)**, **C3 (Ableton, Logic)**	**C4 (Traktor, Resolume)**
Engine Status (section **Status**)	Read-only live MIDI engine status.	—	—

Timecode (Pro)

Header: **Timecode Settings**. Holds the settings for chasing external timecode and for generating MTC.

The Input / Chase section is not active in this version. Its settings are shown and saved, but ATTACCA 1.0 does not decode incoming MTC or LTC, so no cue fires from a timecode trigger and the **Incoming** readout stays empty (see *Timecode triggers* in *Control & Integration*). Generating timecode for other devices is done

with a Timecode cue (see *Show Control Cues*).

[Illustration to be added in a future revision]

Setting	What it does	Options / range	Default
Enable timecode input (chase) (section **Input / Chase**)	Follow external timecode and fire timecode-triggered cues.	on / off	Off
Sync Source	Where timecode comes from: **Mtc** (MIDI Timecode from a MIDI device) or **Ltc** (Linear Timecode decoded from an audio input).	**Mtc**, **Ltc**	**Mtc**
MIDI Input Device	The MTC source device (visible when the source is MTC).	detected MIDI inputs	none
Audio Input Channel	The 1-based audio input channel carrying LTC (visible when the source is LTC); 0 = not configured.	≥ 0	0
SMPTE Format	Frame-rate interpretation — must match the sender.	**24 fps**, **25 fps**, **29.97 df**, **29.97 ndf**, **30 fps**	**30 fps**
On Start	What happens when timecode starts partway into the show: Skip (fire nothing missed), StartAll (fire everything earlier), RecentMinute / RecentHour (fire cues from the last minute/hour), LookbackTime.	Skip, StartAll, RecentMinute, RecentHour, LookbackTime	Skip
On Stop	What happens when timecode stops.	None, HardPause, HardStop	None
Freewheel Time (s)	Keep chasing this long after the signal drops out before treating it as stopped.	0.0–2.0	0.0
Enable timecode output (generate) (section **Output**)	Generate MTC for other devices to chase.	on / off	Off
Incoming / **Outgoing** / **Status** (section **Timecode Status**)	Read-only live timecode readouts.	—	—

Lighting (Pro)

Header: **Lighting Settings**. Configures DMX output over Art-Net, sACN, and USB, plus the light patch.

Where these live: the output settings on this page (both **Enable** switches, broadcast/multicast/unicast modes and targets, Source Name, Priority, loopback, the **Interface**, Art-Net Net/Subnet/Universe, and the USB-DMX devices) are saved per user on **this computer** and are never read from or written to a workspace file. A fresh install has both protocols **off** until you enable one and click **Save**. **Universe Count**, **Start Universe**, **Merge Mode** and the **Light Patch** travel with the workspace. See *Lighting → Enabling lighting output* for why.

[Illustration to be added in a future revision]

Setting	What it does	Options / range	Default
Interface (section **Network Interface**) + **Refresh**	Which network adapter to use for Art-Net and sACN output. Applies to both protocols. If your show network is segregated, pick the specific NIC rather than leaving this on the OS default.	detected adapters	none (OS default)
Enable Art-Net output	Output DMX over Art-Net. Saved per machine.	on / off	Off
Broadcast mode	Broadcast Art-Net to the whole subnet; turn off to unicast to one node.	on / off	On
Target Address	The unicast Art-Net target IP (visible when broadcast is off).	IP address	empty
Art-Net Net	Art-Net 4 Net value.	0–127	0
Art-Net Subnet	Art-Net 4 Subnet value.	0–15	0
Art-Net Universe	Art-Net 4 Universe value (0-indexed per the Art-Net protocol).	0–15	0
Enable sACN output (section **sACN (E1.31)**)	Output DMX over sACN. Saved per machine.	on / off	Off
Source Name	The name ATTACCA advertises in sACN discovery.	text	ATTACCA
Priority (0-200)	sACN packet priority (higher wins at the receiver).	0–200	100
Multicast mode	Multicast sACN (standard); turn off to unicast to one receiver.	on / off	On
Multicast loopback (same-machine monitoring)	Enable to see sACN output in monitoring tools (e.g. sACN View) on this machine. Disable for production.	on / off	On
Unicast Target	The unicast sACN target IP.	IP address	empty
Universe Count (section **DMX Universes**)	Number of DMX universes to output.	1–64	1
Start Universe	First universe number for Art-Net/sACN output.	≥ 1	1
Merge Mode (section **DMX Merge**)	How competing values for the same channel combine: HTP = Highest Takes Precedence, LTP = Latest Takes Precedence.	Htp, Ltp	Ltp
USB-DMX (Enttec DMX USB Pro) section	Output an additional universe through an Enttec DMX USB Pro (FTDI) interface at 40 Hz, alongside Art-Net/sACN. **Add Device** / **Refresh Ports** buttons; each device card has a **Name**, status badge, **Remove** button, editable **COM Port**, **Universe**, and **Enabled** checkbox.	detected COM ports	none

Setting	What it does	Options / range	Default
Light Patch section	A summary of your patch plus **Edit Light Patch...**, which opens the full patch editor inline (fixtures, addresses, groups — see the lighting chapter).	—	—

Broadcast (Pro)

Header: **Broadcast Settings**. Makes ATTACCA *announce* what it is doing to other systems — the mirror image of the OSC/MIDI *input* pages.

[Illustration to be added in a future revision]

Setting	What it does	Options / range	Default
Enable MSC broadcast (section **MSC (MIDI Show Control) Output**)	When enabled, MSC GO is sent when cues fire, MSC ALL_OFF on Panic/Stop All, MSC RESET on Reset All.	on / off	Off
Device ID (0-127)	MSC Device ID to send to; 127 = all-call (send to all devices).	0–127	127
MIDI Patch	MIDI output patch number for MSC messages.	≥ 1	1
Command Format	The MSC device category the messages claim.	**All Types**, **Sound (General)**, **Lighting (General)**, **Machinery (General)**, **Video (General)**, **Projection (General)**, **Process Control**, **Pyro (General)**	**All Types**
Enable OSC broadcast (section **OSC Broadcast**)	Sends OSC events for GO, cue start/stop, panic, stop all, and pause/resume all; external systems subscribe via `/listen` commands and can request a custom `/eventFormat`.	on / off	Off
Destination Host	IP address to send broadcast OSC messages to.	IP address	127.0.0.1
Destination Port	UDP port for broadcast OSC messages.	port	53001

Video (Standard)

Header: **Video Settings**.

[Illustration to be added in a future revision]

Setting	What it does	Options / range	Default
Display (section **Video Output**) + **Refresh**	The default display for video output (individual cues can override it).	detected monitors	none
Width / **Height** (section **Default Display**)	The default stage size in pixels.	320–7680 / 240–4320	1920 × 1080
Presets	One-click stage sizes: **1280×720**, **1920×1080**, **3840×2160**.	—	—
Allow multiple visual cues per display (PIP / overlay mode) (section **Visual Cue Overlap**)	When OFF (default, safe): firing a new visual cue replaces any other visual cue on the same display. When ON: cues on different layers coexist for PIP / overlay effects (max 4 simultaneous visuals per display).	on / off	Off
Enable NDI output (section **NDI Output**)	Streams the video output over NDI to other machines on the network.	on / off	Off
Stream Name	The NDI stream name other software will see.	text	ATTACCA Output
Status	Read-only NDI availability — reads "NDI runtime not detected" if the NDI runtime is not installed.	—	—

The page footer carries a useful tip verbatim: if blacks appear grey on your display, check your GPU settings — NVIDIA Control Panel → Display → Change Resolution → Output dynamic range → set to **Full** (not Limited). This is common with HDMI connections.

Network (Standard)

Header: **Web Remote Monitor** — this page configures the built-in web server that serves a real-time show monitoring page to any browser on the network (see *Running the Show* for using the remote itself).

[Illustration to be added in a future revision]

Setting	What it does	Options / range	Default
Enable Web Remote	Starts the embedded web server for remote monitoring.	on / off	Off
Allow remote (network) access	OFF (default): the Web Remote is reachable only from this machine (localhost) and no firewall rule is created. ON: binds to the selected interface and opens a firewall rule so other devices on the network can connect.	on / off	Off
Port	HTTP port for the web remote server.	1024–65535	8080
Network Interface	Which network interface to bind the Web Remote server to.	detected adapters	all interfaces

Setting	What it does	Options / range	Default
Require Password for View Mode	When ON, the read-only `/view` page also requires the Web Remote password. Control Mode always requires it.	on / off	Off
Password / **Confirm**	The Web Remote password — **independent of the Show Mode password**. Stored as a salted hash in the workspace.	—	empty
Enable Control Mode (section **Remote Control**)	Allow remote browsers to fire cues via `/control`. A Web Remote password is required to save with Control Mode enabled; all commands are rate-limited and logged.	on / off	Off
Status box	Read-only — "Web Remote is disabled", or the URL to open when running.	—	—

Changes on this page take effect when you click **Save**.

Displays

Header: **Display Mapping** — maps logical display numbers to physical monitors. Cues reference display numbers; change the mapping here when switching venues.

[Illustration to be added in a future revision]

Control	What it does
Display	Read-only logical number ("Display 1", "Display 2"…) — this is what Video and Camera cues refer to.
Name	A friendly name for the display, shown in cue inspector dropdowns (e.g. "Projector SL").
Physical Monitor	The physical monitor this display maps to. Each physical monitor may be the target of **one** mapping only (see below).
Fullscreen	Open fullscreen on this monitor; uncheck for a movable window (useful on single-monitor setups so the cue list stays accessible). Default: on.
Standby Black	Holds a black backer on this display for the whole session, so the desktop never shows between cues — and panic fades video out to black rather than back to your wallpaper. Requires a Fullscreen mapping on a multi-monitor rig; on any other row it is silently skipped (the log says why). Saved in the workspace, so it travels to the venue. Default: off.
Test Pattern	Puts an alignment card on this display **immediately** — see below. Never saved; clears on app restart. Same Fullscreen + multi-monitor requirement.
Custom region (px) + **X / Y / W / H** + **Full display**	Confines this display's output to a sub-rectangle of the monitor, in physical pixels (see below). **Full display** clears the region.
+ Add Display	Appends one new row, named "Display N", assigned to the next detected monitor if there is one, with Fullscreen on. Existing rows are never altered.

Control	What it does
- Remove Display	Removes the **last** row (minimum 1). It is not a remove-selected button.

[Illustration to be added in a future revision]

> **Why it works this way:** cues never store a physical monitor — only a logical display number. When you move to a different venue, you reassign the physical monitors here once, and every cue follows. No cue editing needed.

Standby Black

A show display showing the Windows desktop between cues is a classic embarrassment. Check **Standby Black** on a display's row and ATTACCA keeps a black window on that monitor from the moment the workspace loads until the app closes. Cue output always draws over it, so nothing about playback changes; what changes is what the audience sees when nothing is playing, including after a panic — video fades to **black**, not to your desktop.

The backer is deliberately inert: clicking it, or on a touch monitor touching it, does nothing at all and **never takes keyboard focus away from the cue list**. GO, `Space`, and `Esc` keep working no matter where you click.

To clear the backers entirely (at the end of the night, or to get to the desktop on that monitor), press `Ctrl+Shift+F12` — the global "close every output window" hotkey closes backers and test cards along with video windows.

Test Pattern

Check **Test Pattern** on a row and an alignment card appears on that display right away — no Save, no cue. The card carries a 100-pixel grid (heavier lines every 500), a centered crosshair and circle, right-angle corner markers, a one-pixel edge border, and an info chip reading the display number and name, the device, the pixel size, the region if one is set, and the DPI scale. Line it up against your screen, wall, or LED processor, then uncheck it.

Two things to know:

- **It is never saved and never survives a restart** — it exists only for as long as the app is running. It also acts on the mapping as last **saved**, so if you have just edited a row, Save before ticking Test Pattern.
- **A video cue covers the card while it plays, and the card reappears between cues.** So an alignment card forgotten before the house opens is visible to the audience. ATTACCA warns you about exactly this with an amber banner across the Displays page while any card is up: **TEST PATTERN ACTIVE** — *"one or more displays are showing the alignment card. Video cues will cover it while they play, but it reappears between cues. Turn it off before the show."*

[Illustration to be added in a future revision]

Custom output region

By default a display outputs to the whole monitor. Check **Custom region (px)** and the output is confined to a rectangle you specify in physical pixels — everything outside it stays black. This is the LED-wall control: a processor that presents itself to Windows as a 1920×1080 monitor but drives a 1408×832 wall takes a region of 1408 × 832 at 0, 0, and the rest of the signal goes black.

- The X/Y/W/H boxes are physical pixels, relative to that display. Checking the box with empty sizes prefills the monitor's full resolution as a starting point.
- **Full display** clears the region and returns the output to the whole monitor.
- The test pattern renders **inside** the region, so its edge border is exactly the tool for aligning the region to the wall.
- Save refuses zero or negative sizes — *"Display N: custom region width and height must be at least 1 pixel."* and *"Display N: custom region X and Y cannot be negative."* A region larger than the monitor is not refused: it is clamped to the monitor's bounds, the fields are rewritten to the clamped numbers, and a warning goes to the log.

One mapping per physical display

Each physical monitor can be the target of **at most one** display mapping. Point two rows at the same monitor and Save is blocked with:

> "…" is targeted by more than one mapping. Each physical display can have one output mapping. Multiple regions per display is planned for a future update.

Retarget or remove one of the rows. Rows with no monitor assigned are exempt, so half-built mappings never trip it.

Keyboard Shortcuts

Header: **Keyboard Shortcuts**. Every one of the 47 shortcut actions is remappable here — the full default table is in *Reference: Shortcuts, Files & Formats*.

[Illustration to be added in a future revision]

- The page's own instructions: "Click a binding to remap. Press the new key combination, or press Escape to cancel. Enter, Tab, and arrow keys are reserved for core navigation and cannot be reassigned. Changes apply on Save."
- **Search:** filters the list by action name.
- Each row shows the action's display name, its current binding (click it, then press the new keys), the built-in default, and a per-row **Reset** button. **Reset All to Defaults** at the bottom restores everything (it confirms first: "Reset every keyboard shortcut to its built-in default? This cannot be undone.").
- If the chord you press is already assigned, ATTACCA asks whether to reassign it (caption **Reassign Shortcut**) — the other action loses its binding.
- On **Save**, your shortcuts are written per-machine to `%APPDATA%\ATTACCA\shortcuts.json`.

Show Timer Settings

The Show Timer has its own settings dialog, reached **only by right-clicking the Show Timer window** and choosing **Timer Settings…** — there is no gear button and no Preferences page for it. The dialog is titled **Show Timer Settings**, every change applies **live** (no OK/Cancel — just a single **Close** button), and everything here is saved in the workspace.

[Illustration to be added in a future revision]

Display section:

Setting	What it does	Default
Always on top	Keeps the timer window above other windows.	On
Fullscreen	Fills the timer's monitor (this checkbox and the right-click menu are the *only* ways in and out of fullscreen — `Esc` does not exit).	Off
Show controls	Shows the Elapsed/Countdown/Message control strip at the bottom of the timer window.	On
Detach controls into separate window	Moves the control strip into a floating **Timer Controls** window (e.g. controls at the desk, display on stage).	Off
Show seconds	Show seconds on the current-time clock.	On
Time format	**24-hour (HH:MM:SS)** or **12-hour (h:MM:SS AM/PM)**. Also drives the live clock on the Preferences General page.	24-hour (HH:MM:SS)
Display windows	How many timer windows to open: 1, 2, 3, or 4. Windows 2–4 are display-only mirrors titled "Show Timer (2)" etc.	1

Visible sections — checkboxes for **Current time**, **Elapsed timer**, **Countdown**, and **Next cue info**; all on by default.

Appearance sections — each display element has its own color and font size:

Section	Fields	Defaults
Background	Color, Opacity slider (0–100%)	near-black, fully opaque
Current Time	Color, Font size	white, 72
Elapsed Timer	Color, Font size, Label color	blue, 40, grey label
Countdown	Color, Font size, Label color, Label font size, **Flash on zero**, Flash color	40, label 18, flash on, red flash
Next Cue Info	Color, Font size	light grey, 18
Message Display	Color, Font size, Background	white, 48, transparent

Font-size boxes are editable combos with presets from 10 to 200.

License (View menu)

License activation is not in Preferences: open **View → License…** (or **View → About ATTACCA… → Manage License…**) for the **ATTACCA License** window. It holds the key box and **Activate**, plus **Refresh License** and **Deactivate This Machine** once a license is active (and **Upgrade to Pro…** while the license is Standard), and displays your tier, licensed email, purchase date, and update-plan end date. The About window shows license *status* only. Your license is stored per-machine in `%APPDATA%\ATTACCA`. For the full walk-through — tiers, machine seats, offline behavior, and the update plan — see *Licensing & Activation*.

[Illustration to be added in a future revision]

Common issues

- **Save is greyed out** — nothing has changed yet. It enables as soon as any setting differs from its saved value.
- **A settings page shows the upgrade dialog instead of opening** — Video and Network need Standard; OSC, MIDI, Timecode, Lighting, and Broadcast need Pro. Enter a key via **View → License…**.
- **Changed the latency mode but nothing happened** — latency changes take effect on the next app restart (the page says so in a yellow note).
- **Clicked Cancel but my audio device change stuck** — device switching is intentionally immediate and is the one change Cancel does not revert; switch back in the device list if needed.
- **Set OSC up but my console can’t fire cues** — first check the server is enabled at all (it is off by default); then note that without a passcode, remote OSC clients are read-only by design. Set a **Passcode** on the OSC page and have the client authenticate with `/connect`.
- **My settings didn’t follow me to another computer** — most settings live in the workspace and do travel; keyboard shortcuts, the update-check toggle, and your license are per-machine.
- **Can’t find the Show Timer settings** — right-click the Show Timer window itself and choose **Timer Settings…**.

Reference: Shortcuts, Files & Formats

This chapter collects the tables you will want taped to the desk: every default keyboard shortcut, every file ATTACCA reads or writes, the media formats it plays, the network ports it uses, and how updates work.

Default keyboard shortcuts

All 47 actions below are remappable in **Preferences → Keyboard Shortcuts** (see *Settings Reference*). Your custom bindings are stored per-machine in `%APPDATA%\ATTACCA\shortcuts.json`; a workspace can additionally carry its own overrides inside the `.atca` file, and those win over your per-machine bindings when that workspace is open.

[Illustration to be added in a future revision]

Transport & playhead

Action	Default key
GO	`Space`
Panic	`Esc`
Stop Selected Cue	`S`
Pause / Resume Selected Cue	`P`
Pause All	`[`
Resume All	`]`
Move Playhead Up	`Alt+Up`
Move Playhead Down	`Alt+Down`

Editing

Action	Default key
Undo	`Ctrl+Z`
Redo	`Ctrl+Y`
Cut	`Ctrl+X`
Copy	`Ctrl+C`
Paste	`Ctrl+V`
Duplicate	`Ctrl+D`
Delete	`Del`

Action	Default key
Select All	`Ctrl+A`
Arm / Disarm Selected Cue	`Ctrl+Shift+A`
Toggle Flagged	`F`
Cycle Continue Mode	`C`
Copy Effects	`Ctrl+Shift+C`
Paste Effects	`Ctrl+Shift+V`
Group Selected Cues	`Ctrl+G`
Expand All Groups	`.`
Collapse All Groups	`,`

Cue creation (`Ctrl+0` – `9`)

`Ctrl+2` and `Ctrl+8` are intentionally unassigned in this release (they belong to the disabled Mic and Network cue types).

Action	Default key
New Group Cue	`Ctrl+0`
New Audio Cue	`Ctrl+1`
New Video Cue	`Ctrl+3`
New Camera Cue	`Ctrl+4`
New Text Cue	`Ctrl+5`
New Light Cue	`Ctrl+6`
New Fade Cue	`Ctrl+7`
New MIDI Cue	`Ctrl+9`

File & workspace

Action	Default key
New Workspace	`Ctrl+N`
Open Workspace…	`Ctrl+O`
Save Workspace	`Ctrl+S`
Save Workspace As…	`Ctrl+Shift+S`
Export Show Report…	`Ctrl+Shift+E`

Modes, view & windows

Action	Default key
Enter Show Mode	`Ctrl+Shift+]`
Enter Edit Mode	`Ctrl+Shift+[`
Toggle Show / Edit Mode	`Ctrl+E`
Toggle Inspector	`Ctrl+I`
Toggle Show Timer	`Ctrl+T`
Open Preferences…	`Ctrl+,`
Open Renumber Cues…	`Ctrl+R`

Search

Action	Default key
Search Cues	`Ctrl+F`

Shortcuts that still work while you are typing

Most shortcuts deliberately go quiet while a text field has focus, so typing a cue name never fires a cue. A small set still works mid-typing: **New Workspace, Open Workspace…, Save Workspace, Save Workspace As…, Export Show Report…, Undo, Redo, Search Cues, Open Preferences…, Open Renumber Cues…, Enter Show Mode, Enter Edit Mode, Toggle Show / Edit Mode, Toggle Inspector, and Toggle Show Timer.**

`Esc` (Panic) is **not** among them — pressing `Esc` while your cursor is in a text field does not panic the show. Click out of the field (or press GO's row) first. See *Running the Show* for the full panic behavior.

Reserved and fixed keys

- **Reserved for navigation, never assignable:** `Enter` and `Tab` (with any modifier), and the bare arrow keys, `PageUp` / `PageDown`, `Home` / `End`. Trying to bind one is refused with a message.
- **Fixed, not remappable and not shown in the shortcuts list:**
  - `Ctrl+Shift+F12` — the emergency **kill outputs** hotkey. It is registered with Windows itself, so it closes all video output windows even if the ATTACCA window has stopped responding.
  - **Double-** `Esc` — a second panic within 1 second of the first escalates the fade-stop to an immediate hard stop of everything (see *Running the Show*).

File formats & locations

ATTACCA's per-machine data lives under `%APPDATA%\ATTACCA` (paste that into the Explorer address bar to jump there). Show data lives wherever you save your workspace.

What	Where	Notes
Workspace	`YourShow.atca` (wherever you save it)	Your entire show. Plaintext JSON — see below. Saves are atomic: written to a `.tmp` file first, then swapped in, so a crash mid-save cannot corrupt the existing file. Files over 256 MB refuse to load (safety cap).
Autosave	`YourShow.atca.autosave`, next to the workspace	Written on the autosave interval (default 1 minute; configurable or disabled in Preferences → General). If it is newer than the workspace on open, ATTACCA offers to recover from it.
Backups	`%APPDATA%\ATTACCA\Backups\<name>_<id>\`	A timestamped copy is taken every time you save over an existing workspace; the **10 most recent** are kept per workspace. (Older versions kept backups in a `<name>_backups` folder next to the workspace — those are still read.)
DMX recordings	`recordings\cue_<number>_<timestamp>.dmxrec`, next to the workspace	Binary DMX capture attached to a Light cue. Files over 1 GB refuse to load (safety cap).
License	`%APPDATA%\ATTACCA`	Your activated license and its offline proof of purchase. Use **Deactivate This Machine** in **View → License…** to release the seat properly — see *Licensing & Activation*.
App settings	`%APPDATA%\ATTACCA\settings.json`	Small per-machine flags. Almost everything you set in Preferences is stored in the workspace instead.
Keyboard shortcuts	`%APPDATA%\ATTACCA\shortcuts.json`	Your per-machine remaps; workspace-specific overrides live in the workspace file itself.
Logs	`%APPDATA%\ATTACCA\logs\attacca-YYYYMMDD.log`	One log file per day. The path is shown read-only in Preferences → General → Logging. Attach the current day's log to bug reports.
Crash reports	`%APPDATA%\ATTACCA\crashes\crash_*.json` and `%APPDATA%\ATTACCA\crash.txt`	Diagnostic files written around a crash. `crash.txt` also tells the next launch a crash happened — repeated crashes trigger Safe Mode.
Freeze minidumps	`%APPDATA%\ATTACCA\freeze_*.dmp`	If the UI ever freezes, a small diagnostic dump is written automatically. Include it with a bug report.

What	Where	Notes
MIDI profiles	`%APPDATA%\ATTACCA\midi-profiles\`	Named MIDI mapping profiles (Save/Load/Export/Import on the Preferences MIDI page).
Recent files	`%APPDATA%\ATTACCA\recent_files.json`	The Welcome window's recent-workspaces list.
Update state	`%APPDATA%\ATTACCA\update_settings.json` and `check-for-updates.txt`	Last check time, any skipped version, and the startup-check toggle.

The `.atca` format

A workspace is a plaintext JSON file: a small envelope (`formatVersion` — currently 1 — plus the `appVersion` and `savedAt` timestamp of the save) wrapped around the workspace body with every cue list, cue, and setting. You can open one in any text editor to inspect it, and it diffs cleanly in version control. A file with a *newer* format version than the app understands refuses to load rather than guessing, and says so plainly: *"The file is not damaged — update ATTACCA to open it."*

Two workspace fields were added for display output and are ignored by older data: `standbyBlack` (per display mapping) and `outputRegion` (`x` , `y` , `width` , `height` , in physical pixels; absent or null = the whole display). The video Fill Style also gained the value `"native"` ; a workspace saved with it will not open in older ATTACCA builds.

[Illustration to be added in a future revision]

[Illustration to be added in a future revision]

Supported media formats

The exact filters offered by each **Browse…** dialog:

Media	File picker offers	Notes
Audio (Audio/Mic cues)	`.wav` `.mp3` `.aiff` `.aif` `.flac` `.ogg` `.wma` `.m4a`	Played by the BASS audio engine. The Audio cue's picker also lists video and MIDI files. Multi-channel files play and display all channels.
Video (Video cues)	`.mp4` `.mov` `.avi` `.mkv` `.wmv` `.webm` `.m4v`	Decoded by ATTACCA's bundled FFmpeg — no codec packs needed. Codecs include H.264, H.265/HEVC, ProRes (incl. 4444), HAP, VP8/VP9, AV1, DNxHD/HR, CineForm, MJPEG, and more; in-file audio (AAC, MP3, PCM, AC-3, Vorbis/Opus/FLAC, ALAC, WMA) plays too. The Video picker also accepts still images (`.png` `.jpg` `.jpeg` `.bmp` `.tiff` `.tif` `.gif`).

Media	File picker offers	Notes
Images (Image cues)	`.png` `.jpg` `.jpeg` `.bmp` `.webp` `.gif` `.tif` `.tiff` `.ico`	Static images with hold and fade in/out.
MIDI files (MIDI File cues)	`.mid` `.midi`	Standard MIDI files played out a MIDI patch at an adjustable rate.

The audio and video chapters of this manual cover playback behavior (rates, loops, geometry, effects) in depth.

Network ports

Port	Protocol	Used for	Where configured
53000	UDP	OSC listen port	Preferences → OSC
53000	TCP	OSC over TCP (SLIP-framed); same number as UDP by default — different protocols, no conflict	Preferences → OSC
53001	UDP	OSC replies (and the default OSC Broadcast destination port)	Preferences → OSC / Broadcast
53535	UDP	Plain-text (non-SLIP) OSC-style connections	Preferences → OSC
8080	TCP (HTTP)	Web Remote monitor/control pages	Preferences → Network
53600	TCP	Collaboration (workspace sharing)	Collaboration window
6454	UDP	Art-Net DMX output (standard Art-Net port)	Preferences → Lighting
5568	UDP	sACN (E1.31) DMX output (standard sACN port)	Preferences → Lighting

All the ATTACCA-specific ports (OSC, Web Remote, Collaboration) are changeable; Art-Net and sACN use their industry-standard ports.

Updates

- **Automatic check:** with **Check for updates on startup** enabled (Preferences → General; on by default), ATTACCA checks its GitHub releases at launch, at most once per 24 hours. You can check any time with **File → Check for Updates...**.
- **The Update Available window** shows the new version, its download size, and the release notes, with three choices: **Update Now** (downloads with a live progress bar, then runs the installer), **Remind Me Later**, and **Skip This Version** (that version stops nagging; the next one will notify again).
- **Updates are verified before they run.** The download must come from the official release host, its size and published SHA-256 checksum must match, and the installer's digital signature must belong to the publisher

(Cody Yeager). If any check fails, the file is deleted and the update is refused — ATTACCA will not launch an installer it cannot verify.

- **Manual fallback:** if automatic installation fails, ATTACCA shows you where the verified installer was downloaded so you can run it yourself; you can also always download the latest `ATTACCA.msi` from the website and install over the top. Your settings, license, and recent files are untouched by an update.

[Illustration to be added in a future revision]

Common issues

- **A shortcut "stopped working"** — check whether your cursor is in a text field; most shortcuts (including `Space` for GO and `Esc` for Panic) are suppressed while typing. Also check Preferences → Keyboard Shortcuts for an accidental remap — **Reset All to Defaults** restores everything.
- **A shortcut behaves differently only in one workspace** — that workspace carries its own shortcut overrides. They are saved inside the `.atca` file and take precedence over your per-machine `shortcuts.json`.
- **Can't find backups next to the show file** — backups now live centrally in `%APPDATA%\ATTACCA\Backups\`, ten per workspace.
- **An update download was "refused"** — a verification check (checksum or signature) failed, which usually means a corrupted download; try again, or download the installer manually from the website.
- **A `.dmxrec` or `.atca` file won't load** — very large files hit deliberate safety caps (1 GB for recordings, 256 MB for workspaces), and workspaces saved by a *newer* ATTACCA version refuse to open in an older one. Update the app, or re-save from the newer version.

Troubleshooting, Known Limitations & FAQ

This chapter is in three parts: the documented limitations of version 1.0 (so you can plan around them instead of discovering them), fixes for the problems operators actually hit, and answers to the questions we hear most.

Known limitations in version 1.0

These are deliberate, documented boundaries of this release — each has a planned follow-up. Knowing them ahead of time is the difference between a footnote and an incident.

Unplugging a display during playback is not detected

If a display is unplugged, sleeps, or renegotiates *while a video output window is live on it*, ATTACCA does not find out. Windows usually relocates the window — which can mean a borderless, always-on-top, monitor-sized window with video still playing lands on **your** operator display, covering the ATTACCA interface. Where Windows doesn't relocate it, the video simply keeps playing offscreen: the audio continues and the cue completes invisibly. Neither case is a crash.

What to do in the moment:

- If a video window has landed on top of your interface: **click into the video window, then press `Esc`** — with focus, it panics everything and fades itself out. Then re-fire the cue.
- If the video went offscreen: stop the cue and **GO** again. Every new GO re-checks the connected monitors and resolves the output against what's actually plugged in right now (falling back by display name, then device, then position, then the first monitor).
- Opening a workspace whose saved default display is missing falls back to the primary display for the session, with a warning in the log — your saved display choice in the workspace is not overwritten.

What to do ahead of time: don't hot-unplug displays mid-show. Seat the cables, tape the adapters, and treat display changes as an Edit Mode activity.

`Esc` does not panic while you're typing in a text field

While the text cursor is in any text field, `Esc` cancels the edit — it does not panic. Click out of the field (or press `Enter` / `Tab`) and `Esc` works normally. Panic from OSC `/panic` or MIDI is unaffected, and two panics within 1 second from any source still escalate to the hard stop. During a show, keep focus on the cue grid. Covered in full in *Running the Show*.

Per-output audio levels are not available yet

The **Levels** tab controls a cue's **Main Level** only; per-output faders and matrix routing are a future update (the tab says so where they would appear). Multi-channel audio **files** play correctly with no downmix — every channel appears in the waveform, the Levels tab shows the file's true channel count, and channels map to speakers per your Windows speaker configuration. What you can't do yet is route or trim individual outputs per cue.

ASIO is not supported

Audio output goes through your selected Windows output device (WASAPI/DirectSound). ASIO devices work fine as ordinary Windows audio devices via their manufacturer's driver — ATTACCA just doesn't address them through the ASIO API.

Mixed-DPI multi-monitor rigs: slight video softness

On rigs where monitors run different Windows scale factors, the video output window is sized using the main window's DPI and stretched by Windows onto the other monitor — geometry stays correct, but the image can be slightly soft on the second display. A per-monitor DPI pass is planned. Workaround for the pixel-critical: run the show displays at the same scale factor as the operator display.

Lighting channel labels refresh on the next universe switch

The fixture-name labels on the lighting channel grid read your patch when the Levels tab opens or when you switch universes. If you edit the patch while the tab is open, the labels catch up the next time you switch universes or reselect the cue. The output itself is always current — only the labels lag.

Show Timer options are on the right-click menu

Not a fault — just the one control surface people search for: all Show Timer options live on the timer window's right-click menu. See *Running the Show*.

Troubleshooting

Installing: SmartScreen and antivirus

ATTACCA is digitally signed, but as a newer application it hasn't built the download history SmartScreen leans on. If Windows shows "Windows protected your PC": click **More info**, confirm the publisher, then **Run anyway**.

[Illustration to be added in a future revision]

Two portable-build behaviors that look like problems but aren't:

- **First launch is slow.** The portable `ATTACCA.exe` self-extracts its runtime to `%TEMP%\.net\` the first time it runs on a machine — a one-time delay of several seconds. Later launches are fast.
- **Antivirus activity on first launch.** Many AV products scan those freshly extracted files in `%TEMP%\.net\`, extending the delay. This is routine reputation checking, not an infection warning. If your AV quarantines the app outright, add an exclusion for the exe (and `%TEMP%\.net\`) or use the MSI install instead.

See *Getting Started* for the full install walkthrough.

No sound

Work down this list — it resolves nearly every silent-cue report:

1. **Is ATTACCA on the right output device? View → Audio Output Device** — the checkmark must be on the interface actually connected to your speakers. Selecting a device takes effect immediately.

2. **Check the Windows volume mixer.** Windows keeps a *per-application* volume: right-click the speaker icon in the system tray → **Open volume mixer**. If ATTACCA's slider is down or muted there, no setting inside ATTACCA can help. Set it to 100% and do your mixing on the console.

[Illustration to be added in a future revision]

3. **Is another app holding the device in exclusive mode?** Some DAWs, conferencing apps, and system-test tools take exclusive control of an audio device, silencing everyone else. Close other audio software (a pre-show habit worth keeping) or disable "Allow applications to take exclusive control" in the device's Windows properties.
4. **Is it just one cue?** Check the cue's Main Level, its Armed state, and whether a broken file target is flagged.

Video won't play

- **Check the format first.** ATTACCA bundles its own decoders and plays a wide range of containers and codecs — see the supported-formats table in the video chapter. If a file is outside that range, re-encode it (H.264 in MP4 is the reliable workhorse). Installing a system codec pack will **not** change what ATTACCA can play — it decodes with its own bundled FFmpeg.
- **Black screen with audio, or degraded playback on one machine only?** The app may be running in Safe Mode (below) or its `ffmpeg` folder may be damaged — the log file names any missing decoder DLLs explicitly. Reinstall to restore them.
- **Blacks look grey on the projector?** That's a GPU output-range setting, not a decode problem — see the note in **Preferences → Video**.

Network features unreachable (OSC, Art-Net/sACN, Web Remote)

- **Is the server even on?** The **OSC server is off by default** — check **Enable OSC Server** in Preferences → OSC and Save before blaming the network. (The Web Remote is likewise off by default, and reachable only from this machine until you check **Allow remote (network) access**.)
- **Firewall.** ATTACCA adds Windows Firewall rules for the Web Remote (when you enable network access) and for OSC (when a passcode is set) — but a third-party firewall or a locked-down machine policy can still block them. Verify the app is allowed through for your network profile (Private vs Public matters).
- **VPNs block multicast.** A VPN client on either machine will typically swallow sACN multicast, discovery traffic, and sometimes all LAN traffic. Disconnect the VPN on the show network.
- **Router AP isolation.** Venue Wi-Fi and guest networks often enable AP/client isolation, which stops devices from seeing each other at all — the Web Remote URL will simply time out from a phone. Use a dedicated show network or a hotspot you control.
- **Right interface?** On multi-NIC machines, check the network-interface selections in **Preferences → Lighting** (Art-Net/sACN) and **Preferences → Network** (Web Remote) so traffic leaves the correct port.
- **Web Remote reachable but read-only?** Control Mode is a separate opt-in with a required password — see the Web Remote coverage in *Control & Integration*.

A workspace won't open: "saved by a newer version"

If you open a file that was saved by a **newer** version of ATTACCA than the one you are running, ATTACCA tells you so plainly rather than trying to guess:

> This file was saved by a newer version of ATTACCA (file format N; this version reads up to format M). The file is not damaged — update ATTACCA to open it.

Take that message literally. **The file is fine.** ATTACCA does not attempt a partial load, does not offer to recover from a backup, and does not write anything to the file — a newer format is not corruption, and treating it as corruption is how a good show file gets replaced with a broken one. Update ATTACCA (or open the file on the machine that saved it) and it opens normally.

This is the usual sign that two machines in a rig are on different builds. Bring them to the same version before tech.

[Illustration to be added in a future revision]

"Pull From Cue is not supported in this version"

Light cue levels can be typed as commands on the Light cue's **Levels** tab. One piece of that syntax — **Pull From Cue**, written `= cue N` (also seen as `#pull`) — is **not supported in this version of ATTACCA**.

Two places you will meet it:

- **Typing it.** The command is rejected as you enter it, with a red validation line under the command box: *"Line n: Pull From Cue ('= cue N') is not supported in this version of ATTACCA — support is planned for a future update."* Nothing is silently accepted and nothing is silently sent as zero — an unsupported command is an error, not a dark light.
- **Opening an older workspace that contains it.** The show **still opens.** The affected Light cue is flagged **broken** in the cue list, with that same sentence as its reason, and the rest of the workspace loads and runs normally. Edit the cue's levels to remove the `= cue N` line and the flag clears.

[Illustration to be added in a future revision]

[Illustration to be added in a future revision]

Display output looks wrong

- **A grid or alignment card appears on my output between cues.** A **Test Pattern** checkbox was left on in Preferences → Displays. The page shows an amber **TEST PATTERN ACTIVE** banner while any card is up — uncheck the box (or restart the app, which clears all cards).
- **My output shows black bars, or the picture is confined to a corner.** That display has a **custom region** set. Open Preferences → Displays and click **Full display** on that row to clear it.
- **Standby Black does nothing.** The option needs a **Fullscreen** mapping on a **multi-monitor** rig; on a windowed mapping or a single-monitor machine it is skipped without complaint (the log records why).

- **Preferences won't save — "…is targeted by more than one mapping".** Two rows on the Displays page point at the same physical monitor, which isn't allowed. Retarget or remove one of them. (Multiple output regions on one display is a planned future feature.)
- **"Native" fill style looks the same as Fit.** It is: Native is present in the list but is not applied by this release's renderer. The value is saved with the cue.
- **Display settings didn't survive reopening the workspace.** Display mappings are saved in the workspace and reload with it. If you are seeing this on an older build, update.

The app won't start normally / Safe Mode

If ATTACCA crashes more than three times within an hour, the next launch enters a reduced **Safe Mode** to break the crash loop. Safe Mode is deliberately quiet — there's no banner — but you'll recognize it by what's different:

- Your last workspace is **not** auto-loaded; you land on the Welcome window instead. (Your workspace is fine — open it manually.)
- Video plays through a basic fallback decoder (reduced format support), camera cues show a placeholder instead of a live feed, Effect cues don't run, and the Web Remote doesn't start.
- The log file records `SAFE MODE ACTIVE` near the top.

Safe Mode lasts for that session only. If the crashes were caused by a specific workspace or file, that's usually visible in the log; if the app now runs stably, simply restart it and it comes back in normal mode. A single crash (without the loop) just makes the next launch start video features defensively — also logged, also temporary.

If the app won't start at all: check `%APPDATA%\ATTACCA\logs` for the freshest log, and try renaming `%APPDATA%\ATTACCA` aside temporarily to rule out a damaged settings file — then report it (below).

Recovering after a crash

ATTACCA autosaves your workspace to a companion file (`YourShow.atca.autosave` , next to the workspace) on a timer — every minute by default. If the app closes uncleanly, the next time you open that workspace you'll see:

> "A more recent autosave was found for this workspace. This may contain changes from a session that didn't save properly. Would you like to recover from the autosave?"

[Illustration to be added in a future revision]

Click **Yes** to load the autosaved state (then **File → Save** to make it permanent), or **No** to load your last explicit save. Beyond autosave, every save-over also drops a timestamped backup under `%APPDATA%\ATTACCA\Backups\` — up to 10 per workspace — so even a bad save has a fallback.

Forgotten Show Mode exit password

If the Show Mode password has been lost, there is a built-in recovery: **hold `Ctrl+Shift+Alt+F12` for a full 5 seconds**. ATTACCA forces the workspace back to Edit Mode without the password (and writes a line in the log noting that it happened). From Edit Mode you can set a new password in **Preferences → General → Show**

Mode Security.

You will never be locked out of your own show. That said — the password exists to keep operators out of Edit Mode, and this page defeats it. If that separation matters in your venue, this is the one page of the manual to keep off the ops desk.

Reporting a problem

Send reports to **showtoolsofficial@gmail.com** (also click-to-copy in **View** → **About ATTACCA...**). The more of the following you attach, the faster the fix:

What	Where	Why it helps
The day's log file	`%APPDATA%\ATTACCA\logs\attacca-YYYYMMDD.log` (one file per day — send the date it happened)	The primary diagnostic: startup checks, device choices, warnings, and errors with timestamps
Crash markers	`crash.txt` and `crash.log` in `%APPDATA%\ATTACCA`	Written when the app goes down hard; captures the failure itself
Freeze snapshots	`freeze_YYYY-MM-DD_HH-mm-ss.dmp` in `%APPDATA%\ATTACCA`	When ATTACCA detects its own interface has frozen, it writes a small memory snapshot (a "minidump") that developers can open to see exactly what every thread was doing at that moment. If a freeze happened, this file is gold — attach it
Your workspace	the `.atca` file	Lets us reproduce with your exact cues (it contains no media, just the show structure)
What you did	a couple of sentences	"Fired cue 12 while cue 4 was fading" beats a hundred guesses

Paste `%APPDATA%\ATTACCA` into the File Explorer address bar to get to all of these in one hop.

[Illustration to be added in a future revision]

Updates

ATTACCA checks for a new release when it starts (at most once every 24 hours; turn this off in **Preferences** → **General** → **Check for updates on startup**), and you can check any time with **File** → **Check for Updates...**. When a new version exists, the **Update Available** window shows the version, download size, and a **What's New** list, with three choices: **Update Now** (downloads with a progress bar, then installs), **Remind Me Later**, or **Skip This Version** (that version won't be offered again).

[Illustration to be added in a future revision]

Every download is verified before it runs — size, checksum, and the digital signature of the installer — and the update installs over your existing copy without touching your workspaces, preferences, or license. You can also always download an installer manually from the website and run it over your install; offline machines update this way.

Pricing note: new versions are covered by the optional **$49/year update plan**. It gates *new* versions only — **the version you have installed keeps working forever**, plan or not, license or not. If a build is newer than your plan, a dismissible amber bar says so (never during a show), and downloads are refused until you renew. There is no monthly fee and nothing expires. See *Licensing & Activation*.

FAQ

Why is ATTACCA louder than my music app playing the same file? Short answer: most consumer players quietly apply loudness normalization; ATTACCA plays your file at its actual level, exactly as it will hit the console. Full explanation in *Audio in Depth*.

Can I move my show folder to another drive or machine? Yes. Workspaces remember media locations relative to the workspace file, so a folder that travels with its media plays untouched on the new machine. If anything can't be found (media that lived outside the folder), the Relink Media dialog appears on load and re-matches everything from one folder you point it at.

Does the free version expire? No. Free is free forever — no time limit, no nag timer, no cue-count decay. Neither do the paid tiers: a purchased license never expires, and letting the optional update plan lapse never takes a feature away.

Do I need the internet? Only to activate, refresh, or deactivate a license — three actions, all of them in Edit Mode. After activation the license is validated locally, forever, and update checks are optional. See "Running offline" in *Running the Show* for the short list of features that need a (local) network.

Can I open my QLab workspace in ATTACCA? Not in version 1.0 — there is no QLab file importer. Shows need to be rebuilt in ATTACCA. Your media files, of course, carry straight over.

Where are my settings and backups stored? In `%APPDATA%\ATTACCA` — preferences and license there, logs in `logs`, and up to 10 timestamped backups per workspace in `Backups`. Show-specific settings (including most Preferences pages) are saved inside the `.atca` workspace file itself, so they travel with the show. Uninstalling the app leaves all of it in place.

Why does my camera's light stay on even when no camera cue is running? By design. ATTACCA opens and holds every camera referenced by your workspace warm from the moment the workspace loads, so a camera cue's first frame appears instantly on GO instead of after the 2–3 second device-open delay. A warm camera's LED stays lit — the light means "ready", not "broadcasting". (While warmed but not visible, frames aren't even decoded.)

Can two operators run the same show? In version 1.0, Collaboration gives a second machine a live *observer* view — it follows the primary's playhead and sees property edits, read-only. It is not a second control position; there is one operator per show. (The Web Remote's Control Mode can give a trusted browser GO/stop/panic — see *Control & Integration*.)

Can I open the same workspace on two machines at once? ATTACCA warns you if a workspace appears to be open in another instance — two machines saving the same file is how shows get eaten. Use Collaboration for a second viewer instead.

Can I change the keyboard shortcuts? Almost all of them — **Preferences → Keyboard Shortcuts**. The exceptions are the safety-critical ones: `Esc` is always Panic, and `Ctrl+Shift+F12` (close all video outputs) is registered with Windows and fixed.

Licensing & Activation

ATTACCA ships in three tiers — **Free**, **Standard**, and **Pro** — and one optional extra, the update plan. This chapter is the complete reference: what each tier unlocks, how to enter a key, how machines are managed, what happens offline, and what changes when a license is removed. *Getting Started* has the short version; this is the full one.

Two promises frame everything below, and neither has an asterisk:

- **A license never expires.** Free is free forever, and a purchased tier stays purchased.
- **A license change never interrupts a running show.** Nothing in this chapter can stop a cue that is already playing.

The three tiers

Free is a complete audio show-control rig: unlimited cues and cue lists, the full editor, groups and carts, triggers, undo/redo, Show Mode, the Show Timer, and per-cue audio effects. No time limit, no cue cap, no nag timer.

Standard adds the **video family** — Video, Camera, Image, and Text cues, with display mapping, geometry, and the output windows — plus **audio markers** and the **Web Remote**.

Pro adds the show network: **lighting** (Art-Net, sACN, USB-DMX, the Light Patch, Light Dashboard, DMX Status, DMX recording), **MIDI** (triggers, MIDI Learn, MSC, MIDI File and Timecode cues), **OSC** control, **Lua scripting**, **serial and HTTP** cues, **collaboration**, and **broadcast** output.

What each tier unlocks

> **Generated, not transcribed.** The two tables below are printed straight out of the enforcement code by the `DumpEnforcedMatrix` test (`tests/CueFlow.Tests/Licensing/TierMatrixContractTests.cs`). If a tier ever moves in the code, regenerate this table rather than editing it by hand — the code is the contract, and two earlier hand-written matrices had already drifted from it.

Cue types — the minimum tier that can create and fire each type:

Tier	Cue types
Free	Arm, Audio, Devamp*, Disarm, Fade, GoTo, Group, Load, Memo, Mic*, Pause, Reset, Start, Stop, Target*, Timer, Wait
Standard	Camera, Image, Text, Video
Pro	Effect, Http, Light, Midi, MidiFile, Network*, Osc, Script, Serial, Timecode

* Mic, Devamp, Target, and Network are **not available in this version** — their toolbar buttons are disabled regardless of tier. They are listed here because the tier table covers every cue type the file format knows about.

Modules — the minimum tier for each feature area:

Tier	Modules
Free	Audio, AudioEffects
Standard	AudioMarkers, Camera, Video, WebRemote
Pro	Broadcast, Collaboration, Lighting, MIDI, Network, OSC, Scripting, Timecode

Per-output audio levels and matrix routing appear in neither table because they do not exist yet in any tier — see *Troubleshooting, Known Limitations & FAQ*.

Pricing

Tier	Price	Notes
Free	$0	Never expires
Standard	$199 one-time	
Pro	$399 one-time	
Standard → Pro upgrade	$200	Your existing key becomes Pro — no new key is issued
Update plan	$49/year	Optional — only needed for **new** versions. The version you own keeps working forever

There is no monthly subscription.

What a locked cue looks like

Cue types above your tier are never hidden and never deleted. They sit in the cue list **dimmed to half opacity with a grey padlock glyph** in the status column, and the padlock's tooltip names the tier that unlocks that specific cue — *This cue requires ATTACCA Standard* or *This cue requires ATTACCA Pro*.

A locked cue can be selected and read, but it will not fire: GO passes over it, and a Start cue aimed at one is refused with a line in the log.

> *[Illustration to be added in a future revision]*

Clicking the toolbar button for a cue type above your tier opens a dialog explaining the feature, with **Enter License Key**, **Upgrade…**, and **Close** buttons. Pasting a locked cue type is refused the same way. **Upgrade…** opens the upgrade window described under *Upgrading Standard → Pro* below — it reads your license state and routes you to the right place, whether that is the upgrade checkout, the pricing page, or the License window.

> **Why it works this way:** opening a Pro show on a Free machine must never hide or delete anything. Your file is not modified by a tier change — video cues, markers, patch data, and lighting levels all survive intact and come straight back when you activate.

Activating a license

1. Open **View → License…** (also reachable from **View → About ATTACCA… → Manage License…**, and from the **Enter License Key** button on the upgrade dialog). The window is titled **ATTACCA License**.
2. Paste your key exactly as it was sent. Keys are long — paste the whole thing. Surrounding spaces and line breaks are ignored.
3. Click **Activate**.

[Illustration to be added in a future revision]

ATTACCA contacts the license server, registers this computer against the license, and shows the result. On success the window displays your **tier**, **Licensed to** (the purchase email), **Purchased** (the purchase date), and **Updates until** (the end of your update plan).

The new tier takes effect **immediately** — the cue list repaints in place, locked cues unlock where they sit, and nothing needs restarting or scrolling.

[Illustration to be added in a future revision]

Machine seats

A license covers a limited number of computers at once. If you try to activate one machine too many, the window shows **"All machine seats are in use"** and lists the machines currently holding a seat, with the date each was activated:

> This license is already activated on the machines below. Deactivate one to free a seat, then activation will retry automatically.

Select a machine, deactivate it, and activation of the computer in front of you retries by itself. The machine you freed reverts to Free the next time it re-checks the license online — its stored license keeps working until then.

[Illustration to be added in a future revision]

Refresh License — picking up an upgrade

Refresh License re-validates the license online and re-reads your tier. This is how a **Standard → Pro upgrade** bought on the website appears in the app: no new key is issued — the key you already have simply becomes Pro. Refresh License picks it up, and so does any app restart with an internet connection.

Upgrading Standard → Pro

You can start the upgrade from inside the app: click **Upgrade to Pro…** on the License window (shown while your license is Standard), or **Upgrade…** on the tier-gate dialog. ATTACCA first checks your license online, then opens the checkout page in your browser with your key already filled in — the upgrade is purchased on the website, against the key you already have.

Starting the checkout needs an internet connection: if the license server can't be reached, ATTACCA says so and stops rather than sending an unverified key to checkout. Your license is unaffected.

After completing the purchase, click **I've completed my upgrade — Refresh License** in the same window (or **Refresh License** on the License window). Your tier updates in place — locked cues unlock where they sit, nothing needs restarting. If the app still says Standard right after the purchase, the upgrade can take a moment to land on the license server; try **Refresh License** again shortly.

Deactivating this machine

Deactivate This Machine frees this computer's seat so the license can move to a new machine. It needs an internet connection: if the server can't be reached, the seat is **not** freed and ATTACCA says so rather than pretending.

After deactivation, ATTACCA drops to the Free tier. Cues above Free re-dim immediately — but see the next section for what that does *not* do.

What a downgrade does not do

This is the part that matters at 7:58 PM.

- **Running cues keep running.** Deactivating a license, letting an update plan lapse, or dropping a tier never stops a cue that is already playing. Tier enforcement refuses to *start* things; it never stops them.
- **Nothing is deleted.** Locked cues stay in the file with all their settings.
- **Nothing is saved differently.** A workspace saved on Free still carries its video cues, markers, and lighting data.

The visible effect of a downgrade is exactly this: cues above your new tier dim and get a padlock, and the affected toolbar buttons start opening the upgrade dialog again.

Running offline

Once activated, your license is stored on the computer with a cryptographic proof of purchase, and **ATTACCA never needs the internet again**:

- Starting the app offline, opening a workspace, running the whole show, GO, panic — all fully licensed, with no nags and no grace period counting down.
- Restarting offline is fine. The stored proof is what the app reads at launch; a failed network check is simply ignored, not treated as an expired license.
- At startup ATTACCA quietly re-checks the license in the background **when** the internet happens to be available. That background check is what picks up a plan upgrade or a seat you freed from another

machine.

Exactly four actions need a connection: **activating**, **refreshing**, **deactivating**, and **starting a Standard → Pro upgrade** (the license is verified before checkout opens).

> **Why it works this way:** a show machine on a locked-down venue network, or no network at all, is the normal case — not the exception. A license that phones home is a license that can fail at half-hour.

Your update plan

Buying a tier buys that version of ATTACCA forever. The optional **$49/year update plan** buys *newer* versions, and the **Updates until** date on the License window is where you check it.

- **An expired plan never disables anything.** Your tier does not drop, no feature turns off, and the version you have installed keeps working in full, forever.
- What an expired plan does: newer builds are not offered, and downloading one is refused with *"Your update plan has ended. Renew it to install newer versions of ATTACCA."*
- If you are running a build that is newer than your plan (for example after moving a license onto a machine that already had a later version installed), a **dismissible amber bar** appears across the top of the window explaining that this build postdates your plan, with **Renew…** and **Dismiss** buttons.
- **The bar never appears during a show.** It is suppressed in Show Mode and while any cue is running.
- **If you just renewed:** open **View → License…** and click **Refresh License**.

[Illustration to be added in a future revision]

Beta licenses (ended)

The beta program ended on **September 1, 2026**. Beta keys (`ATTACCA-…`) no longer activate — entering one is refused and nothing is stored — and a beta key left on a machine from the beta period now loads as **Free**. Nothing in your workspace files is affected: cues above Free simply dim with a padlock until a purchased license is activated (see *Activating a license*).

Common issues

- **"This license key was not found…"** — the key was mistyped or only partly pasted. Re-copy the *entire* key from your purchase email; they are long by design.
- **"All machine seats are in use"** — the license is already active on its full complement of computers. Deactivate one from the list shown, or from the License window on that computer.
- **"This license has been suspended"** — the license was disabled by support (for example after a refund). Contact support.
- **"This license has expired"** — a time-limited license ran out. Perpetual licenses do not expire; contact support.

- **"Could not reach the license server…"** — no internet, or a firewall in the way. Activation needs a connection once. An already-activated license is completely unaffected by this.
- **"This machine is no longer activated…"** — the seat was freed from another computer. Enter the key again to re-activate.
- **I upgraded on the website but the app still says Standard** — click **Refresh License** on the License window (or restart with internet). No new key is issued for an upgrade.
- **A cue is dimmed with a padlock** — its type needs a higher tier; hover the padlock to see which one.
- **My key looks absurdly long** — normal. Keys are cryptographically signed to prevent tampering.