



AS OF

0.4.0

MARITIME LIGHTHOUSE

SEPTEMBER 2026

Roadmap

Where Lighthouse is going after 0.4.0 — and the order we plan to get there in. Built for our own crews first, so this list is shaped by shows, not by a marketing calendar.

Lighthouse started as a way to run our pixel bars from one Mac without a console. 0.4.0 made it a good citizen of a real show network: a grandMA3 can own it, a Stream Deck can fire it, and cues fade. The next stretch is about running it from more places and on more hardware, without the app growing into something it is not.

HOW TO READ THIS

Shipped is in the download today. **Next** is being built for the next release. **Later** is committed but not scheduled. **Considering** is on the table and depends on what our crews and yours ask for. Nothing here is a promise of a date; everything here is a thing we actually intend to do.

Control and show operation

SHIPPED

Console, Companion, crossfades — 0.4.0

sACN input with an 11-channel fixture profile and GDTF; a Bitfocus Companion module with actions, feedbacks, variables and presets; per-cue fade times with true pixel crossfades; a Control API with an event stream and an optional token.

NEXT

OSC

The same control model over OSC, so QLab, TouchOSC, Ableton and any show-control system that speaks it can recall cues, set levels and blackout without touching HTTP. Same verbs as the API: `/lighthouse/cue` , `/lighthouse/preset` , `/lighthouse/master` , `/lighthouse/blackout` . Feedback over OSC for TouchOSC layouts.

Cue list

Saved cues in an ordered list with a Go button, follow times and per-step fades, so a simple show runs from the keyboard or a single Companion button without a console. The buttons you have today stay; the list is a view over them.

Companion module in the Bitfocus store

Submitted to Bitfocus so it installs from Companion's own module list instead of an imported package. Verified against Companion 4 and 5.

LATER

Timecode and MIDI

MIDI Show Control and note triggers for cues; LTC / MTC timecode chase for the cue list. Enough for a corporate open or a museum loop; not a full timeline editor.

Backup Mac

Two Macs running the same rig file, the second at a lower sACN priority with mirrored control state, so a crash mid-show hands over without a dark stage. The priority half already works today; the state mirroring is the work.

Looks and the engine

NEXT

Look layering

Run two looks at once — a slow Ocean under a Sparkle — with a blend mode and a level per layer. This is what most "can it do..." requests turn out to be.

Per-fixture and per-group looks

Fixture groups drawn on the Design page, and cues that address a group instead of the whole rig, so the header bars can run something different from the floor run.

LATER

Custom looks

A small, safe scripting surface so a programmer can write a look — a function of pixel position, time and the parameters — save it with the rig and share it. The built-in looks would be written in the same language.

Media on the strip

Play an image or a short video along the run as a colour source, for logo washes and pre-rendered effects.

Fixtures and controllers

NEXT

RGBW and generic strip

Four-channel pixels end to end, and a generic bar type where you set the length and pixel count, so rigs that are not Pixel Bar Pro can be drawn, wired and patched the same way.

LATER

More controllers

Profiles for Advatek, Falcon and generic 8- and 16-port boxes with their own per-port pixel and universe budgets, so the Wire page can warn you before the show does.

Fixture library import

Add a bar or a connector to the library from a small definition file, and share libraries between rigs and crews.

Platforms

The biggest structural change on the list. Lighthouse today is a macOS app where the engine and the window are one process. Everything below starts with splitting them.

NEXT

Headless core

The engine and the control model become a service; the window becomes a web page the service serves. On a Mac nothing changes — it still opens as an app — but the same page then opens in a browser on any device on the show LAN. This is the step that makes everything else on this page possible.

LATER

iPad

The Cues and Live pages in Safari on an iPad, talking to the core on the Mac — a wireless monitor and cue panel, with no app store in the way. A native iPad engine only if browsers turn out not to be enough.

Windows and Linux

The same app on the other two desktops, signed for Windows. Same rig files, same looks, same Companion module.

Raspberry Pi show appliance

An SD-card image that boots into the headless core and runs the last rig on power-up — a permanent install in a bar or a lobby with no computer to babysit. Programmed from a Mac or an iPad over the LAN; driven from a console or Companion like any other Lighthouse.

Considering

IDEAS

- **Art-Net input** alongside sACN input, for consoles that only speak Art-Net.
- **Multi-rig** — several rig files open at once, each on its own universes, from one window.
- **Pixel mapping from the drawing** — looks that use the bar's real position on the canvas (a wipe that crosses the stage left to right regardless of cable order).
- **Show logging** — a timestamped record of every cue and who fired it, for post-show notes.
- **Live audio reactivity** — level and beat from an input, as a parameter looks can use.

TELL US WHAT TO MOVE UP

If one of these would change how you run a show, say so at getstage.work. The order above is ours; it moves when a real rig needs it to.

Maritime Lighthouse is built by Maritime Creative, St. Augustine, Florida. Roadmap as of Lighthouse 0.4.0, September 2026. Everything listed is intent, not a commitment to a date.