

Showbook user guide

Showbook is a **desktop show file library for video switchers**. It keeps every show for your Barco Event Master (E2, S3-4K, EX, Encore3) and Analog Way LivePremier (Aquilon) — and reads Midra 4K and Alta 4K — with a version history; it draws the patch, the screens, the layers and the presets; it writes PDF documentation, test patterns, multiviewer layouts and Bitfocus Companion pages; it converts a show between platforms with a report of what carried, what was adapted and what was dropped; and it pulls from and pushes to the live hardware.

Before you rely on this: every driver was built against the vendors' own simulators — Barco's Event Master Toolset simulator (E2 9.2, Encore3 10.0) and Analog Way's LivePremier, Midra 4K and Alta 4K simulators — and their published protocol guides. The parsers, the library, the history, the conversion, the PDF and the Companion export are covered by tests on captures from those simulators, and the desktop app has pulled a show from the LivePremier simulator and written the PDF for it.

But **nothing here has been run against a physical switcher**, the Barco simulators do not serve the JSON-RPC API so the Event Master live path has never answered a request, **no Event Master preset or cue file has ever been seen** (the simulators never had one saved), and an `.awc` has been uploaded to a simulator but never *applied* — that step reboots the device. See [What has been checked](#) before trusting a pull, a push or a preset read from a backup. **Status: in development.**

This codebase was created with AI assistance, directed and reviewed by a human author.

What a show is

Showbook keeps one brand-neutral model and reads each vendor's own file into it. A show has:

- a **platform** and a **model** — which family of switcher it belongs to (Event Master, LivePremier, Midra 4K, Alta 4K, LiveCore, PDS-4K, PixelHue, or generic) and which frame;
- the **system**: firmware, device name, native rate, genlock, and the **chassis** — every frame, slot, card and connector;
- **inputs** and **outputs**, each on a connector with a format; **sources** built on inputs, stills, multiviewers or auxes, with an area of interest;
- **screens** (and auxiliary screens): a canvas size, the outputs tiled onto it, the layers it has and their capacity, and the transition;
- **presets** — for each screen they target, the background and every layer's source, rectangle, crop, opacity and border — and **master presets** that recall one preset per screen;

- **cues**, **multiviewers** with their layouts and windows, and **stills**;
- the **vendor files** the show came from (an Event Master backup archive, a LivePremier `.awc`), kept opaque and byte-exact; and **notes** — everything the importer could not place, so nothing is silently dropped.

Each vendor's spelling is kept where it matters: an Event Master preset is `pre:<id>` in the frame's own numbering, an Aquilon memory is `pre:<slot>`, layers are `layer:<n>`, and a LivePremier screen remembers which preset letter was on program when it was captured.

The library

A library is a folder of plain files:

<code>shows/<id>/show.json</code>	the current version of each show
<code>shows/<id>/history/index.json</code>	its version list
<code>shows/<id>/history/<sha256>.json.gz</code>	every saved version, content-addressed
<code>vendor/<sha256><ext></code>	the vendor files, once each

Nothing is a database and nothing is hidden. Copy the folder and you have copied the library.

Settings → Library → Folder chooses it. The default is `Showbook` in your Documents folder. Put it inside a folder that Dropbox, OneDrive or Google Drive's desktop client carries and the whole library follows you between machines — or use the [Sync](#) page to mirror it over the drives' own APIs.

Settings → Your name is recorded on every version you save.

Settings themselves live in `settings.json` in the app's configuration folder — on macOS `~/Library/Application Support/com.allansargeant.showbook`, on Windows `%APPDATA%\com.allansargeant.showbook`, on Linux `~/.config/com.allansargeant.showbook`. The path is shown under **Settings → About**.

Getting a show in

Import a file

Library → Import file... takes:

File	From
E3Backup.tar.gz , a backup .zip	Event Master Toolset: Configuration → Backup, or the Encore3's /api/backup
settings.xml	the frame's xml/ folder, bare
.awc	LivePremier: Web RCS → Setup → Backup/Restore → export
.json , a saved device store	LivePremier Web RCS GET /api/stores/device , saved to disk
.json , Showbook's own	another Showbook, or Documents & export → Showbook JSON

An Event Master archive is unpacked and settings.xml is read with everything beside it — presets/ , cues/ , userkey/ , hwconfig.xml on an Encore3 — and the archive itself is kept as a vendor file. An .awc is read for its manifest (the device, its modules, the firmware) and kept whole; its payload is encrypted and Showbook does not look inside it. The show's configuration comes from the device store, not from the .awc — pull it from the device, or import a saved store.

Import a folder

Library → Import folder... takes the frame's xml/ folder (the one holding settings.xml) or any folder that holds it, such as an unpacked backup.

Pull from a device

See [Devices](#). A pull creates a new show, or updates an existing one and records a version.

Start from nothing

Library → New show asks for a name, a platform and a model, and opens an empty show. Add inputs, outputs, screens and layers on the Patch and Screens tabs; the chassis fills in when the show is first pulled from a device or converted from one that has one.

The show window

The title is editable. Beside it: the platform, the model and the firmware. On the right:

- a **commit message** and **Save version**. Every save is a version — see [History](#). The button reads *Saved* when nothing has changed. Enter in the message field saves too.
- **Duplicate** copies the show, with a new name, into a new entry with no history.
- **Delete** removes the show and its whole history from the library. It asks first, and there is no undo.
- **Close** returns to the library and drops unsaved edits — save a version first. The sidebar shows a dot beside the show while it has unsaved edits.

The tabs:

Overview

System — platform, model (a list of the models Showbook knows for that platform), firmware, device name, native rate, genlock; tags; free-text show notes.

At a glance — the counts: inputs, outputs, screens, aux, presets, masters, cues, multiviewers, stills, mixer layers.

Chassis — every frame, slot and card, and every connector with what is on it: the input or output it carries, its label, its format, and which screen or multiviewer it feeds. Empty connectors are drawn empty. The PDF prints the same facts as a table per frame.

Patch

Three tables — **Inputs**, **Outputs**, **Sources** — with the label, the connector, the format and, for outputs, the role (screen, aux, multiviewer, test) and what the output feeds. Labels and formats are editable; type a format as `1920x1080p60` (or `1920x1080i50`, `3840x2160p59.94`). A source shows what it is built on (an input, a still, a multiviewer) and its area of interest.

Screens

The list on the left holds every screen and aux. For the selected one: label, canvas width and height, kind, transition time; the **outputs on this screen** with each output's rectangle on the canvas; the **layers** the screen has, with their kind (mixer, background, DSK) and capacity (SL, DL, 4K). Add and remove screens, outputs and layers here.

The canvas on the right draws the screen to scale with its outputs and, on top, the layers of one **state**:

- *Program as captured* (and, on a LivePremier, *Preview as captured*, with the preset letter that was at that end of the T-bar) — what the device was showing when the show was pulled or imported. Read-only.
- any **preset** that targets this screen — editable. Drag a layer to move it, drag the corner handle to resize it. The table under the canvas has the same rectangles as numbers, plus source, visibility, opacity and border, and the background.

A screen that has no captured state and is targeted by no preset draws nothing; pick a preset on the Presets tab and add the screen to it.

Presets

Presets lists every preset (Event Master) or screen memory (Analog Way) with the screens it targets. For the selected one: number, label, notes, and the targets — one block per screen, each with its background and its layers. Add a screen to a preset, remove one, or edit the layers here or on the Screens canvas.

Master presets are recalled as a set: one preset per screen. On an Event Master a preset that targets several destinations is shown as one preset; on a LivePremier a master memory is one of these.

Recall on a device. When a device of the show's platform is known (see [Devices](#)), the selected preset can be recalled on it: **Recall on** recalls it on every screen it targets, **Recall →** on one; **to program** puts it straight on air instead of into preview; **TAKE** transitions every screen. A master preset has its own **Recall**. Each of these writes to a live device and says so in the toast.

Multiviewers

Every multiviewer output, with its layouts. For the selected multiviewer: label and output. For each layout: label, size, and the windows. **Make grid** fills a layout with an even grid of windows; **Set active** marks the layout the device should show. Click a window on the canvas to edit its **source** and **UMD label**; drag to move it, the corner handle to resize.

Cues

An Event Master keeps cues on the frame and they come in with the backup. A LivePremier has no cue list — its sequencing lives in Companion or a timeline tool. A cue is a list of steps: *Recall preset*, *Recall master*, *Take*, *Wait*, *Other*, each with a delay in milliseconds and, where it applies, the preset or master it recalls. Cues are documented in the PDF and exported to Companion; they are not played from Showbook.

Documents & export

PDF documentation — cover; the chassis and the patch tables; one page per screen with its outputs and its captured layers; every preset drawn to scale; multiviewer layouts; cues; a glossary of what the settings mean; the notes the importer left; and the version history. Tick what to include, name who it is prepared for, **Save PDF...** Unsaved edits are included; the history table shows saved versions only.

Test patterns — one PNG per output at the output's own raster, labelled with the output, its connector, the screen it belongs to and its position on the canvas, with arrows pointing at its neighbours; and one PNG per screen showing the whole canvas with every output's region. Three patterns: *Alignment* (grid, centre, safe areas), *Grid*, and *75% bars over grid*. **Render PNGs to a folder...** and play them from a media server or a laptop to prove the patch before content arrives.

Companion page — see [Companion](#).

Showbook JSON — the show as one file, for another Showbook or for a script.

Vendor files

The files the show came from, byte-exact: an Event Master backup archive, a LivePremier `.awc`, a saved device store. Each is listed with its kind, size, hash and when it was captured, and can be exported. Restoring an Event Master backup onto a frame is done from the Event Master Toolset (Configuration → Restore) with the archive exported from here.

History

Every version, newest first, with its message, author, time and a summary. Select one to see what changed against the version before it — added, removed and changed entries, keyed by what they

are. **Restore** makes any version the current one; the restore is itself a new version, so nothing is lost.

Devices

Devices holds the switchers Showbook can talk to. Add one with a name, its platform and its address:

- **Event Master** — the frame's IP. The JSON-RPC API is on port 9999.
- **LivePremier, Midra 4K, Alta 4K** — the Web RCS host, with a port if it is not 80 (the simulators use 3000, 3010 and 3021). AWJ is assumed on port 10606 of the same host.

Probe asks the device who it is and shows the answer.

Pull the show from the device reads the running configuration into a new show, or into an existing show of the same platform as a new version.

- *keep the vendor file* also downloads the `.awc` (LivePremier) or the backup archive (Encore3) and keeps it with the show.
- *deep capture memory layers* (Analog Way) recalls every memory into the preview of every screen in turn to read its layers, because the device store carries a memory's name and not its content. **This writes to the device's preview buffers.** It asks before it starts; do not run it during a show.

Push a show to the device — LivePremier, Midra 4K and Alta 4K only:

- *labels* writes the screen, input, output and memory labels over AWJ.
- *restore the .awc* uploads the show's `.awc`, has the device extract it, and applies it. **The device reboots.** The push asks before it starts, and the log under the button shows each step.

An Event Master has no API for writing configuration. Recall presets and cues from the Presets tab; restore a backup archive from the Event Master Toolset.

Convert

Convert makes a show for a different switcher. Pick the source show, the target platform and the model. **Preview the report** shows what would happen without writing anything; **Convert and save** creates the converted show in the library, with the report kept in its notes; **Open the converted show** takes you to it.

The panel on the right shows what the target model holds — screens, auxes, inputs, outputs, layers in 4K-equivalents, multiviewers — and what the show needs. The report has three levels:

- **carried** — came across as it was;

- **adapted** — came across in the nearest form the target has: an Event Master preset that targets three destinations becomes three screen memories and a master memory on a LivePremier, and three memories with a master become one multi-destination preset going the other way; layer capacities are re-costed (a 4K layer is two DL or four SL); ids are re-spelled in the target's own numbering; test patterns are reset because the pattern names differ;
- **dropped** — has no home on the target: screens beyond the model's count, layers beyond its capacity, multiviewers it does not have, the source vendor file.

The chassis, cards and connectors are the target model's. Inputs and outputs keep their labels and formats and need patching to the target's connectors on the Patch tab.

Sync

Sync mirrors the library with somewhere else. It is a two-way mirror that never deletes: a file that is newer on one side is copied to the other, and a file missing on one side is copied there.

- **A folder** — any folder: the Dropbox, OneDrive or Google Drive desktop client's folder, a NAS, a USB stick. No account needed.
- **Dropbox / Google Drive / OneDrive (API)** — over the drive's own API, for a machine without the desktop client. Each needs an app registration of your own — a Dropbox App Console app, a Google Cloud OAuth client of type *Desktop*, or a Microsoft Entra app with a public client and the loopback redirect — whose client id (and, for Google, client secret) go in the fields. **Connect** opens the provider's sign-in page in your browser and waits for it to return to `http://127.0.0.1:<port>/callback`. Tokens are kept in the settings file; **Refresh token** renews them when the provider's expire.

Sync both ways, **Push only** and **Pull only** run it, and the toast reports what moved.

Companion

Documents & export → **Companion page** writes a Bitfocus Companion page: a TAKE per screen and one button per preset (and master memory on Analog Way, cue on Event Master), wired to the `barco-eventmaster` or `anaLogway-awj` module. Fill in the connection label and the device address the module will use, tick what to include, and **Export page**....

In Companion: **Import / Export** → **Import**, choose the file, pick the page to put it on. Companion offers to create the connection the page names or to map it onto one you already have; set its address on the Connections page if you did not fill it in here.

Read a page back... reads a page exported from Companion and checks which of its buttons still match this show — a preset that was renumbered, a cue that no longer exists — so a page built for last year's show can be checked against this year's.

What has been checked

Part	Checked against	Not yet
Event Master store, backup archives	Encore3 10.0.2 and E2 9.2 simulator stores; a packed <code>E3Backup.tar.gz</code> round trip	a real frame's backup; any preset or cue file
Event Master live (JSON-RPC, <code>/api/backup</code>)	Barco's API guide and the Bitfocus module's use of it	a request answered by a frame — the simulators do not serve the API
LivePremier store, REST, AWJ, <code>.awc</code>	LivePremier simulator 6.2.73: store, REST recalls, <code>.awc</code> download → upload → extract, AWJ reads and writes	a physical Aquilon; apply of an <code>.awc</code> (reboots); deep capture
Midra 4K / Alta 4K	Pulse 4K 3.2.29 and Zenith 200 1.3.7 simulator stores, read live	writes and REST recalls on either
Conversion, Companion export, PDF, test patterns, library, history, folder sync	unit tests; PDF and pull from the running desktop app	a page imported into a running Companion
Dropbox, Google Drive, OneDrive	the providers' API references	a live account

Running it from source

```
npm install
npm run app          # tauri dev
npm run app:build    # release bundle for this platform
```

`npm run dev` alone serves the front end to a browser tab with two simulator captures in memory — every screen works, but files and devices need the desktop app.

Showbook · v0.1.0 · guide updated 2026-09-19 · stoa-works-labs.com/software/showbook/

Latest version of this guide: stoa-works-labs.com/software/showbook/guide/ · Source and issues:

<https://github.com/stoa-works-labs/showbook>