

AV Central

Setup and Integration Guide — hosting the dashboard, connecting every service API, and pushing it live to the whole team.

VERSION

**1.1.0 Beta
52**

UPDATED

**August 8,
2026**

DEFAULT ADDRESS

http://localhost:4400

ADMIN ACCESS

**Signed-in
admin**

WHAT'S INSIDE**1****Before you start**

What you need — hardware and software

3**Host it permanently**

Auto-start, static IP, screens around the building

5**Connect the APIs**

Planning Center, ProPresenter, BoxCast, and more

7**Who can do what**

Viewer / Editor / Admin, trusted machines, hidden pages

9**Updates, backups, troubleshooting**

Keeping Sunday safe

2**Install and first run**

Up and running in five minutes

4**Switch from demo to live**

How the draft → push-live flow works

6**AV Central sign-on**

Who you are, what it grants, and what replaced the PIN

8**Configure the room**

Roles, mic assignments, devices, alerts

10**Quick reference**

The one-page cheat sheet for FOH

1 Before you start

The dashboard is a self-contained web app. One computer runs it; everyone else — FOH, the video booth, the stage display TV, your phone — just opens a browser. There is nothing to install on any other machine.

You need

ITEM	DETAILS
A host computer	Any always-on Mac, Windows, or Linux machine on the production network. A Mac mini or small NUC is perfect. The ProPresenter Mac works too — the dashboard is very light.
Node.js 18 or newer	Free from nodejs.org (choose the LTS installer). This is the only software requirement — the app has zero dependencies, no <code>npm install</code> , and no build step.
The app folder	<code>calvary-techarts-dashboard.zip</code> , unzipped somewhere permanent, e.g. <code>/Applications/techarts-dashboard</code> or <code>C:\techarts-dashboard</code> .
Network access	The host must reach your production VLAN (consoles, receivers, ProPresenter) and the internet (Planning Center, BoxCast, YouTube, Facebook). If audio/video gear lives on an isolated VLAN, give the host a leg on both.

- ✓ **Demo mode means you can set this up on a Tuesday.** The app boots with a fully simulated service — plan advancing, SPL moving, batteries draining — so you can arrange every screen and train the team before a single credential is entered.

2 Install and first run

- 1 Install Node.js from nodejs.org (LTS). Confirm it in a terminal: `node --version` should print v18 or higher.
- 2 Unzip the install zip (`calvary-techarts-dashboard-v1.1.0-beta.52-install.zip`) to its permanent home.
- 3 Open a terminal in that folder and start it. The simplest first run is:
`node server/index.js`
 That launches the dashboard directly — nothing to install, no build step. Once you know it works, switch to `npm start` (below), which runs the same server under an auto-restart watcher that reboots itself after crashes **and** after software updates, so you never reload the backend by hand. (`npm run start:once` is the bare `node server/index.js` behavior if you ever want it under npm.)
- 4 Open `http://localhost:4400` in a browser. The demo service is already live.
- 5 Sign in as the administrator. **There is no shared PIN any more** — the dashboard was built around one for its first year and it has been retired. Admin is now a named person or an approved machine (§6, §7). On a brand-new install the first account you create owns the whole system; make it, sign in, then open Admin → General and press **Push live** once so the live settings and the draft agree.

! If you have a note anywhere that says "the Admin PIN is ...", throw it away. Nothing in the dashboard asks for it. Every place that used to prompt for a PIN now either lets a signed-in admin straight through or refuses with a sentence telling you to sign in — it never shows a PIN box that cannot open the door. **The one PIN that survives is the parent-paging PIN**, which is a different thing entirely and is set in Admin → Screens & Displays → Paging.

Take a lap around the app

PAGE	URL	USE IT FOR
Dashboard	<code>/</code>	Role-based widget grids. Pick a Position from the chip in the top bar; press the  button next to it to drag, resize, add, or remove widgets. Editing a layout needs Editor or better; anyone holding it is never asked for anything (§7). Since Beta 39 that same chip is also where an Admin finds the Screens group — the seven link-only wall pages, listed below a dividing line inside the one dropdown.
Mics	<code>/mics</code>	Who is on which mic and pack, with live battery and RF. Packs show the name programmed on the receiver itself.
Status	<code>/status</code>	Every local device (pinged every 20 seconds while healthy, less often once one has been down a while) grouped by room, plus vendor status pages. Click any device for its full record; click a switch for its port map, which states how old the reading is.
Reports	<code>/reports</code>	Post-service analytics: loudest and quietest moments, viewership, scheduled vs actual timing, incidents.
Trends	<code>/trends</code>	Week-over-week history, filterable by service time, plus the SPL-compliance and streaming-SLA reports.

Stage	<code>/stage</code>	Full-screen confidence display (link-only — not in the nav): clock, countdown, current and next item, producer messages. The card that configures it is Admin → Screens & Displays → Stage display; you position and size the panels right on the page (sign in as an editor or admin, then  Edit layout). Producer notifications overlay every panel and auto-clear after 2 minutes (configurable).
Paging	<code>/paging</code>	Parent paging (codes → the side screens via ProPresenter) and guest-services room counts, in one page with tabs.
Admin	<code>/admin</code>	Everything configurable — behind Admin , whether access control is switched on or not (§7). The version that's actually running shows next to the page title.

i **New in Beta 39 — the top bar reports the state of the service, on every dashboard page.** Four states, worst first: **ALERT** (a critical alert is active and unacknowledged), **WARNING** (a warning is active), **LIVE** (a service is running), **STANDBY** (nothing running). Each one changes the header's colour *and* shows its word — the word is the carrier, because Path Green and Amber are all but indistinguishable to a red-green colourblind operator and two of the four states would otherwise be one colour. The colour still changes with motion switched off. Wall screens and the seven link-only kiosk pages deliberately get none of it: a pulsing red header on a lobby TV is the Beta 19 lesson relearned.

i Add `?tv=1` to any URL for wall displays, e.g. `http://10.0.5.20:4400/status?tv=1`. TV mode stretches edge-to-edge, packs the Status page into 3–4 columns so the whole network fits without scrolling, and dims itself overnight (schedule in Admin → General). It is a **display flag only** — it grants no access, and `/admin?tv=1` is still gated (§7).

3 Host it permanently

› Give the host a fixed address

Reserve an IP for the host machine in your router or DHCP server (for example 10.0.5.20). Every screen in the building will use it, so it must never change. If your network has local DNS, add a friendly name like `techarts.local`.

› Start automatically — macOS (launchd)

Create `~/Library/LaunchAgents/org.calvarychurch.techarts.plist`:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
  "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0"><dict>
  <key>Label</key><string>org.calvarychurch.techarts</string>
  <key>ProgramArguments</key><array>
    <string>/usr/local/bin/node</string>
    <string>/Applications/techarts-dashboard/server/watch.js</string>
  </array>
  <key>RunAtLoad</key><true/>
  <key>KeepAlive</key><true/>
</dict></plist>
```

Then load it once: `launchctl load ~/Library/LaunchAgents/org.calvarychurch.techarts.plist`. It now survives reboots and restarts itself if it ever crashes. Pointing it at `server/watch.js` (the same thing `npm start` runs) also means remote software updates apply themselves. (Run `which node` to confirm the node path — on Apple Silicon it is often `/opt/homebrew/bin/node`.)

› Start automatically — Windows (Task Scheduler)

- 1 Open Task Scheduler → Create Task. Name it TechArts Dashboard.
- 2 General tab: check **Run whether user is logged on or not**.
- 3 Triggers: New → Begin the task **At startup**.
- 4 Actions: New → Program: `C:\Program Files\nodejs\node.exe` · Arguments: `server\watch.js` · Start in: `C:\techarts-dashboard`.
- 5 Settings: check **Restart the task if it fails**.

› Start automatically — Linux (systemd)

```
# /etc/systemd/system/techarts.service
[Unit]
Description=AV Central Service Dashboard
After=network.target

[Service]
ExecStart=/usr/bin/node /opt/techarts-dashboard/server/watch.js
Restart=always
```

```
User=techarts

[Install]
WantedBy=multi-user.target
```

Enable with `sudo systemctl enable --now techarts`.

› Put it on every screen

- **FOH and booth computers:** bookmark `http://10.0.5.20:4400` and pick each operator's Position once — it is remembered per browser.
- **Stage display TV:** point any TV stick, Apple TV browser, or a mini PC in kiosk mode at `/stage`. Chrome kiosk example: `chrome --kiosk http://10.0.5.20:4400/stage`.
- **Status TV in the booth:** `/status?tv=1` — every device on one screen, no scrolling.
- **Signage / lobby TVs:** `/signage`, and **give each one a name in its own address** — `/signage?screen=lobby`, `/signage?screen=atrium`. From Beta 19 that name is what a Stream Deck button or the on-screen arrows control, so two TVs with names can be held on different pages independently. A screen opened at plain `/signage` is the one called `default` — which means **two unnamed TVs are clones and move together**. Use plain lowercase names with no spaces; anything else gets squeezed down to fit. **New in Beta 22:** Admin → Screens & Displays → Hidden pages builds these addresses for you, with a name box and a copy button per screen — see §7.
- **Phones and tablets:** same address — layouts collapse to a single column automatically. Add to Home Screen for a full-screen feel.

› Optional: secure access from home

The dashboard is built for your internal network. If staff need it off-site, install Tailscale (free for small teams) on the host and on staff devices — they can then reach the same address from anywhere with zero firewall changes. If you prefer a public URL, put a reverse proxy such as Caddy in front for automatic HTTPS, and add authentication at the proxy. Never port-forward it raw.

! Get your admin accounts and your booth machines sorted before the first Sunday. Viewing is open to anyone on the network. Everything that changes something is behind Editor or Admin (§7), and there is no shared secret that hands either of them out — it is a signed-in person, or a machine you have approved. Approve at least two machines at Admin level, on two different computers, before you rely on this: an install with one approved machine and no reachable admin account can lock itself out of its own Admin page.

4 Switch from demo to live

Everything in Admin edits a **draft**. Dashboards keep running on the last published config until you press **Push live**, which updates every connected screen instantly — mid-service safe.

- 1 Open Admin → Integrations and set up each connector you use (section 5). Press **Test connection** on each — you want a green check before Sunday.
- 2 In Admin → General, switch **Mode** from Demo to Live.
- 3 Press **Push live**. The simulator stops and the connectors take over, feeding the exact same widgets.

✓ You can flip back to Demo any time — handy for training new volunteers on a weeknight without touching the live systems.

i **Credentials stay on the host.** API keys are stored in `server/data/config.json` on the host machine and are never sent to browsers.

5 Connect the APIs

Each connector below maps to a card in **Admin** → **Integrations**. Enable it, fill in the fields shown, press **Test connection**, then **Push live**. One integration is **not** on this tab and has a section of its own: **AV Central sign-on** lives under Admin → General, because it is a credential rather than a data feed — see §6. **Trusted machines** and the access-control switch are covered in §7, and since **Beta 40** they are not on the Admin page at all: they moved to the **AV Central Settings page**.

Planning Center Services

CLOUD API

Drives the service plan widget and team data for mic assignments — **and the service clock itself**: when the plan goes live in PCO the service report starts recording, and when LIVE ends the report archives. (See Admin → General → Service reports for when a report is kept.)

- 1 Sign in at api.planningcenteronline.com/oauth/applications with an account that can view Services and control LIVE.
- 2 Under Personal Access Tokens, choose New Token, name it TechArts Dashboard.
- 3 Copy the Application ID and Secret it generates.
- 4 Find your Service Type ID: open your Sunday service type in Services — the number in the address bar after `/service_types/` is the ID.

Fields: Application ID · Secret · Service Type ID · Drive service clock (default on)

ProPresenter 7 (GFX 1 — Side Screens · GFX 2 — Video Wall · GFX 3 — 411 Room)

LOCAL NETWORK

All three machines are monitored, each with its own GFX widget (live slide thumbnail + full slide list). GFX 1 feeds the message clocks, and receives stage-display messages and parent-paging codes. **Changed in Beta 17:** the Cue Sheet's slide block for an item now comes from whichever GFX machine that item names in **Admin** → **Links**, read directly from that machine — see §8. Presentation names no longer have to match item titles for slides to appear.

- 1 On each ProPresenter Mac: Settings → Network → check **Enable Network**. Note the port (default 1025).
- 2 Give each Mac a reserved IP so it never moves.

Fields: Instances (JSON list: id, label, host, port) · Message name for stage texts ("Stage")

i **Superseded in Beta 17.** The old name-matching behaviour — the dashboard resolving a link's ProPresenter playlist name against the console's playlist tree — is gone. It had two fuzzy string matches in series, and when either missed, the Cue Sheet's slide block was simply *absent* with nothing to explain why. Items now name a machine instead; nothing depends on names lining up.

i **Beta 14:** the earlier "Auto-advance LIVE from ProPresenter" toggle has been removed — ProPresenter firing a linked presentation no longer steps PCO Services LIVE forward on its own. Advance the plan in Planning Center as usual; the Cue Sheet pull-in above still works independently of this.

ProPresenter Stage Display (video countdown)

LOCAL NETWORK

NEW IN BETA 17

Shows the time remaining on whatever video ProPresenter is currently playing, in a **Video Countdown** widget. This is a **separate connection** from the ProPresenter entry above, with its own password.

- 1 On the GFX 1 Mac: ProPresenter → Preferences → Network. That one screen is the authority for both things this connector needs — the **Stage Display** password, which is *not* the network/API password, and the port Stage Display is actually listening on. Read the port there rather than assuming it matches the port the ProPresenter connector uses; it commonly is the same, but that is a habit, not a rule.
- 2 In **Admin** → **Integrations** → **ProPresenter Stage**, enter that password, set **Enabled**, and press **Test**. Leave **Host** blank to follow GFX 1's host, or set it explicitly.
- 3 Add the **Video Countdown** widget to whichever role should see it — Admin → Roles, or the ⓘ button in the dashboard header, then "+ Add widget" and search **Video Countdown**. **It is not on any role by default.**

Fields: Enabled · Host (blank = follow GFX 1) · Stage Display port, read from ProPresenter → Preferences → Network (the field starts at 1025) · Password (masked)

! **Read this before relying on it.** ProPresenter's supported API has no endpoint for a video's live playback position, so this connector uses the older **Stage Display** channel, which Renewed Vision does not document and can change without warning. It is the only connection in the dashboard that works this way. It is built to fail quietly rather than loudly — if the protocol changes, the widget says it has no countdown rather than breaking anything else — but it carries no guarantee, and a ProPresenter update could stop it working. Everything else in this guide talks a supported interface.

i **Where the port comes from, and why the password isn't shared.** The Stage Display port is set in **ProPresenter** → **Preferences** → **Network** — read it there. It is commonly the same port the ProPresenter API connector uses, and the field here starts at 1025 for that reason, but a default is a starting value, not a promise: if Test says nothing is listening on the Stage Display channel, that screen is where you check. The password and the port are kept as this connector's own settings deliberately — sharing one field with the API connector would break silently the day one changes and not the other, and would make a Stage Display password problem look like an API failure. Not every clip type reports a countdown — when nothing is counting, the widget says **No countdown active** rather than going blank.

BoxCast

CLOUD API

Embeds the live player and reads broadcast status and viewer metrics. BoxCast can also drive the service lifecycle as a **fallback** when Planning Center isn't driving it (the "Drive service" toggle).

- 1 In the BoxCast dashboard, open Settings → Developers (ask your BoxCast admin if you don't see it).
- 2 Create an API client — copy the Client ID and Client Secret.
- 3 Your Channel ID is in the embed code for your channel (the string after `boxcast.tv/channel/`).

Fields: Client ID · Client Secret · Channel ID · Broadcast ID (optional override) · Drive service (fallback)

YouTube Live

CLOUD API

Live concurrent viewers for the Stream Stats widget and reports, plus the live-chat tab in the chat widget.

- 1 Go to console.cloud.google.com → create a project (e.g. AV Central).
- 2 APIs and Services → Library → enable **YouTube Data API v3**.
- 3 Credentials → Create credentials → API key. Restrict it to the YouTube Data API.
- 4 Your Channel ID (starts with UC...) is under YouTube Studio → Settings → Channel → Advanced.

Fields: API Key · Channel ID · Read live chat

Facebook Live

CLOUD API

Live viewer counts from the church page.

- 1 At developers.facebook.com, create an app (type: Business) tied to the church page.
- 2 In Graph API Explorer, generate a Page access token with `pages_read_engagement` and `live-video read` permissions, then exchange it for a long-lived token (Tools → Access Token Debugger → Extend).
- 3 Your Page ID is on the page's About screen.

Fields: Page ID · Page Access Token

! Long-lived page tokens still expire eventually. If Facebook numbers stop, regenerate the token — two minutes of work.

Church Online Platform

CLOUD (VIA PROXY)

ChOP does not publish a public stats API, so the connector accepts any HTTPS endpoint that returns JSON with a current-attendance number — a tiny proxy you host, a Lambda, or a Google Apps Script that reads your ChOP admin metrics.

Fields: Metrics endpoint URL · Bearer token (optional) · Attendance field path (dot notation, e.g. `currentAttendance`) · public + host chat embed URLs

Smaart — SPL

LOCAL NETWORK

Feeds the SPL meter, the high-SPL alert, and the loudness timeline in every report. This connector speaks Smaart's real **WebSocket API (v3)** — the same one the Smaart WebSocket API Client uses — not a web page or a REST URL.

- i **Supported (API v3):** full Smaart **v8.3.0 – v9.0.1**, Smaart Di 2.1+, and Smaart SPL. The SPL/metric stream (what we read) is served by **full Smaart** and **Smaart SPL** — Smaart Di doesn't expose calibrated inputs. Smaart 7 (API v1) and Smaart 8.0–8.2 (API v2) use a different, older protocol this connector can't read; update Smaart if you're on one of those.

- 1 On the Smaart machine: **Options** → **API** → enable the API server. Note the **port** shown there (the example in Rational's docs is 26000; use whatever yours shows).
- 2 Run a **calibrated SPL input** in Smaart (a calibrated measurement mic / SPL logging) for the SPL meter. For the live **RTA**, also have a **spectrum measurement active** in Smaart — the RTA bars stream from it.
- 3 In the dashboard, enter the machine's IP and the API port, pick the SPL metric you want (default **SPL A Slow**; others include SPL C Slow, SPL Fast...) and the RTA banding (default 1/3 octave), then press **Test connection**. The Test button reports your exact Smaart product and version, whether it's supported, a live SPL reading, and whether an RTA spectrum measurement is active — a definitive verdict in one click.

Fields: Host/IP · API server port · SPL metric · Device / Channel (optional) · RTA spectrum measurement (optional) · RTA banding · RTA scale floor/ceiling dBFS (optional) · **Shows:** live SPL + the full metric set (SPL A/C Slow & Fast, LAeq/LCeq/Leq, Peak C, FS Peak) + the live 1/3-octave RTA

Shure Wireless — Axient Digital, ULX-D, QLX-D

LOCAL NETWORK

Wireless Workbench itself has no public API, so the dashboard talks straight to the receivers using Shure's documented command strings on TCP port 2202 — the same telemetry WWB displays (battery, RF, frequency, channel names). Channel names programmed on the receiver show up automatically as the mic's label everywhere.

- 1 Give each receiver a reserved IP (you likely already have this for WWB).
- 2 List each receiver in the connector's JSON field with a unit name and channel count.
- 3 **Programme a Channel Name on every channel** in Wireless Workbench (or on the receiver front panel). That name — not the label in Mic Assignments — is what the dashboard and the printed assignment sheet display for that pack.

Field (JSON list): [{"host":"10.5.70.22","port":2202,"model":"ULXD40","unit":"ULXD 1","channels":4}]

i If packs show a generic name — **Mic 1** , **Wireless 2** , **Bass Pack** — the cause is at the hardware, not in the dashboard, and there is a one-glance test. Battery, RF and the channel name all arrive in the **same poll**, so look at the Mics page: **batteries and RF showing, but generic names** means the receivers are answering fine and those channels simply have no Channel Name programmed — set them in Wireless Workbench. **Battery and RF blank as well** means the receivers aren't answering the dashboard host at all on **10.5.70.22 - .25** port 2202 — a network path, a firewall, or network control switched off on the receiver. Fix that first and the names follow. The Wireless widget also names any receiver that failed to connect on a "not reachable — check TCP 2202" chip.

i Telemetry is matched to people through Admin → Mic Assignments (unit + channel). PSM900/PSM1000 racks that support command strings report the same way. Live frequencies also feed the RF Coordination widget, which flags carriers parked too close together.

Allen & Heath dLive — FOH and Monitors

LOCAL NETWORK

Both consoles are monitored over MIDI/TCP (port 51325), and channel names sync to the faders straight from Mic Assignments automatically on every Push live.

Fields: Consoles (JSON list: id, label, host, port) · Auto-label faders (checkbox)

ETC Ion — lighting cue preview

LOCAL NETWORK

Read-only OSC over TCP (port 3032). The dashboard subscribes to the console and shows the active cue and the pending cue about to fire. Enable OSC RX/TX in the console Shell → Network settings (OSC 1.0 packet framing).

New in Beta 17 — the whole cue list. The Lighting Cues widget now has a ▼ **Full cue list** toggle that reads the entire cue list off the console with the live cue marked and scrolled into view. This uses ETC's documented OSC Get commands over the same connection and port — nothing new to open or enable. The list is fetched on its own slow schedule and refreshed when the console reports a change, never on the normal poll, because a long show is one request per cue.

Fields: Console IP · OSC TCP port (3032) · Cue list (default **1**)

i **Cue list** only matters if your show is programmed on a list other than 1 — the dashboard also detects it automatically from the cue the console reports as live, so in most cases the default is right. Press **Test** for a definite answer: it distinguishes "**OSC Get supported**" (with the console model and software version) from "**port open but Get unsupported**" from "**unreachable**". On a console that doesn't support it, the widget collapses to today's active/pending view and states the reason rather than going blank.

AJA HELO · HyperDeck ×2 · Ross switcher · OBS · Dante

LOCAL NETWORK

The Broadcast Gear widget shows HELO stream/record state (REST), HyperDeck transport, timecode, and loaded clip (TCP 9993), the Ross RossTalk link (TCP 7788), and OBS scene/stream/record state (obs-websocket v5). The HELO Log widget shows connection status and a dashboard-derived history of streaming/recording state changes. Dante presence (mDNS) flags any expected Dante device that disappears from the network. Fixed IPs are the only setup most of these need.

i **Beta 14:** the HELO Log widget's raw mirror of the encoder's own `/logwatch.tpl` device-log page has been removed. The connection status and status-change history above are unaffected.

Fields: HELO IP · Decks (JSON list) · Switcher IP + port · OBS host/port/password · Dante expected-device list

In-house video feed (NDI, through an MJPEG bridge)

LOCAL NETWORK

With this on, the Livestream widget and the stage display stop using the BoxCast web player and show the in-house program feed instead, a second or two behind the room rather than half a minute. Browsers cannot play NDI, so something on the network has to turn NDI into a picture a browser can draw, and the dashboard reads that.

It has to be MJPEG. It cannot be HLS. Both screens draw this feed into an image, and an image can decode MJPEG and nothing else. Point this at an `.m3u8` and every screen shows a broken picture. The field used to be labelled "MJPEG or HLS", which promised something the dashboard has never been able to display; the label now says MJPEG only, and pressing **Test connection** against a bridge that answers with anything other than MJPEG tells you so at that moment instead of on a Sunday.

The chain that actually runs here

ProPresenter → NDI → **OBS** with the DistroAV plugin (NDI 6) → **OBS Virtual Camera** → **VLC** reading that virtual camera as a DirectShow device → MJPEG over HTTP → the dashboard.

- 1 In OBS, add the NDI source (DistroAV), then **Start Virtual Camera**.
- 2 Start VLC reading the virtual camera and publishing MJPEG over HTTP. On Windows, from a command prompt:

```
vlc dshow:// :dshow-vdev="OBS Virtual Camera" :dshow-adev="none"
--sout "#transcode{vcodec=MJPEG,fps=15}:std{access=http{mime=multipart/x-mixed-replace},mux=mpjpeg,dst=:8080/ndi.mjpg}"
```

- 3 In **Admin** → **Integrations** → **NDI program feed**, enter that address — `http://<that-machine>:8080/ndi.mjpg` — give the feed a label, set how many seconds without a frame counts as frozen (6 is a sensible start), then **Test connection** and **Push live**.

! **Two recipes that used to be recommended here do not work and have been removed.** OBS's `obs-mjpeg` output and the one-line `vlc "ndi://SOURCE"` invocation are both dead ends — VLC reads the OBS virtual camera as a capture device, not from an NDI address. If you find either of them written down anywhere, it is out of date.

✓ **The status light means something now.** It used to be green whenever the connector was switched on — it was reporting the checkbox, not the picture, so a bridge that died mid-service left a frozen frame under a green light and looked perfectly healthy. It now reports whether a frame actually arrived, and says what happened in words underneath when it hasn't. The dashboards and the stage screens read the same indicator, so they cannot disagree about it. **Also new:** the dashboard pulls the feed once and shares it with every screen. Before this, six dashboards and two stage TVs meant eight separate full-rate pulls through one laptop.

Fields: Bridge MJPEG output URL · Feed label · Call the feed frozen after (seconds with no frame) · **Ours:** NDI source `10.5.70.117`, output `DISPLAY-LIVE26 - NDI 1`, mapped to GFX 3 — 411 Room

Subsplash

CLOUD API

Reads current live media status. The Subsplash Platform API is provisioned per account — ask your Subsplash account manager to enable API access and issue a key.

Fields: API Key

Vendor status pages and local devices

NO KEYS NEEDED

These two work with zero credentials. **Vendor status:** any Statuspage-powered page (BoxCast, Planning Center, Subsplash, Church Online, and Cloudflare are preloaded) — add more under Admin → Devices with just the URL. **Local devices:** everything in Admin → Devices is probed every 20 seconds (TCP with ping fallback). A device that stays down past the alert window (default 60 seconds, Admin → Alerts) raises ONE critical alert with the likely cause already diagnosed — and the alert acknowledges itself when the device comes back.

6 AV Central sign-on

This dashboard has never had per-person logins of its own. Identity here is a **production position** picked in a browser and remembered on that screen. **AV Central** — the Production Atlas application — does have real accounts, with per-module access, and it is where the person behind an action comes from. This section connects the two, so that somebody already signed in over there lands here **already set to their position, with no picker**, and, if AV Central marks them an admin on its **Service** module, can do the admin-level work without anything further to type.

The shared PIN is gone. It used to be the independent second path described here, and it was retired during the merge — nothing in the dashboard asks for one now, and no route accepts one. What replaced it is stronger rather than weaker: an action that used to want the PIN now wants a real admin identity, either a signed-in AV Central administrator or a machine you have approved at Admin level (\$7). If you have been reading this section as "the PIN is the fallback", the fallback is now the **Admin trusted machine**, and the advice to keep two of them on two different computers is what makes that fallback real.

i **Two ways the two applications can run, and this section describes one of them.** AV Central and this dashboard can run as **one program**, in which case the identity is simply shared in memory and there is no secret to paste and no hand-off token at all — that is how the merged build runs, and it is what you get out of the box. They can also run as **two separate programs**, in which case the shared secret and the hand-off token below are exactly how a person crosses from one to the other. Everything from here down applies to that second arrangement. The step-by-step account of how the two applications came together is in the per-version notes documents, not in this guide.

› Setting it up — two steps, and one of them is on the other Mac

- 1 On the **AV Central Mac**: generate the shared secret there and copy it. It lives in AV Central's own `server/data/hub.json`. **AV Central never transmits it** — it is carried across by hand, once. Both sides must end up holding the same value or every hand-off is refused.
- 2 On this dashboard: **Admin** → **General** → **AV Central sign-on**. Tick **Enabled**, paste the secret into **Shared secret**, and press **↑ Push live**.

The field never shows the secret back to you — it renders empty and the chip beside it reads **configured** or **not set**. Leaving it blank on a later edit **keeps** what is already stored rather than erasing it, so editing the church name on the same tab can't silently wipe a value that only exists on paper. Use **Clear** when you actually want it gone.

i **The secret never reaches a browser.** It is stripped out of the configuration every screen receives — the same masking the paging PIN has always had — so it is not sitting in the live update frame that the hallway TVs on `/signage` and `/board` are holding open. That is the single most important line in the feature: the secret is what mints admin, so anyone holding it could forge a token and walk straight in as one.

› What it actually does

BEHAVIOUR	DETAIL
Lands you on your position	The position AV Central assigned is written to that screen and the dashboard mounts on it once, before the first render — no picker, no flash of the wrong layout.

The position is a suggestion, not a lock	Deliberate, because the booth is a shared machine: Change position in the header keeps working exactly as it does today. The hand-off is also scrubbed out of the address bar as it is consumed, so a refresh two hours later can't quietly drag somebody back to the position AV Central picked.
Admin, without anything to type	Only for a user AV Central marks as an admin on its Service module . A signed-in non-admin does not satisfy the gate, and is told so in a sentence rather than being offered a box that cannot open the door. With access control on (§7): admin → Admin, edit → Editor, anything else → Viewer.
The hand-off token	Minted by AV Central with a two-minute life , and redeemable exactly once . The dashboard enforces the expiry the token carries and refuses a second use of the same token.
The session it creates	12 hours — a shift, not a day. If it runs out mid-service, click through from AV Central again, or work from a machine you have approved at Editor or Admin level, which needs no session at all.
A refusal is never a dead end	Three plain messages, none of which names a person (this lands on a shared booth screen): <i>couldn't confirm who you are · that link had already been used, or had been sitting too long · sign-on isn't switched on for this dashboard yet</i> . In every case you get a normal, fully usable dashboard and the position picker.
A position AV Central doesn't recognise	If AV Central assigns a position this dashboard doesn't have, the person lands on the first configured position rather than an empty screen, and the swap is written to the device log where Admin can see it — a silent downgrade is exactly the kind of drift that goes unnoticed for months.
Accepted and refused hand-offs are logged	Device log, under AV Central SSO — accepted (with the position and whether it granted admin), refused (with the reason), and any position fallback.

› The position list AV Central reads

Positions on this dashboard are user-definable — add "Broadcast Director" in **Admin** → **Roles & Positions** and it is a real position immediately. **New in Beta 20:** `GET /api/roles/ids` publishes this dashboard's live list (ids and display names only) so AV Central's own picker can read it instead of hard-coding five that were never all real here. It needs no key, because it publishes nothing that `/api/config` doesn't already send to every browser in the building. **Changed in Beta 22:** the list is now **eight** positions — your seven operator positions plus the new **Admin** position — and each entry says whether it is admin-only, so AV Central can still assign Admin while this dashboard keeps it out of a volunteer's picker.

i This one needs a change on the AV Central side to be worth anything. The endpoint exists here and answers now; AV Central still falls back to a hard-coded five until somebody queues that work over there — and it falls back whenever it cannot reach this dashboard, which is what its `serviceDashboardUrl` pointing at `localhost` guarantees. Until both are fixed, a position added here can't be assigned in AV Central, and a position deleted here keeps being assigned — which is the case the fallback above exists to catch. See §7 for the one field that decides it.

! Fixed in Beta 38 — creating an AV Central account with a module switched off now really switches it off. If you create an account over there and set a module to hidden or view-only at the moment you create it, that restriction used to be **silently dropped**: the account came back reporting success, with full access to the module you had just restricted. The only way to find out was to go and look. It is honoured now, and a typo'd module name or level is refused with a message rather than skipped. **Anyone whose account was created that way still has whatever access the drop gave them** — the fix applies to accounts created from now on and does not go back and tighten the ones already made. Open the account list and check the people you meant to restrict.

! Know what you are agreeing to before you switch this on. Turning it on makes this dashboard **trust AV Central's word on who is an admin** — compromise one application and you have both. That is a deliberate choice, and the design keeps **Admin trusted machines** as a completely independent path so that neither one depends on the other — their cookie is signed with a per-install secret of its own, not with the AV Central secret, precisely so that switching AV Central off cannot take the break-glass with it. Be equally clear about the wire: both applications are LAN-local and neither runs HTTPS, so the hand-off token travels over **plain HTTP inside the building**. What limits exposure is the **two-minute expiry** and the **single-use check**, not transport security. If you would rather not extend admin this way, leave **AV Central sign-on** switched off — that is how it ships, and nothing else in the dashboard changes.

i Not yet exercised against a real AV Central. The hand-off was built and tested against tokens minted by the exact code AV Central 0.0.2 Beta 13 ships, and every check passes — but no build machine can reach the AV Central Mac, so nobody has watched a real person click through end to end. Do that once, off-Sunday, before relying on it. The rest of what shipped in Beta 20 is in the **Version History** document.

7 Who can do what — Viewer, Editor, Admin

The dashboard once had two states: anyone on the network could look at everything, and one shared PIN unlocked everything else. There are now three levels in front of the whole app — **Viewer**, **Editor**, **Admin** — checked on the server on every one of the **89** routes that changes anything, and the shared PIN has been retired entirely. None of it is done by hiding buttons: a hidden button is a suggestion, and this is a boundary.

! Read this before anything else in this section: access control ships switched OFF, and switching it off does not mean "no doors". With it off, the levels below stop deciding anything and most of the app is open to anyone on the network — but the **43 actions that carried a credential check before any of this existed still carry one**. Restarting the server, applying an update, changing a trusted machine, taking the platform backup, minting or revoking a QR sticker, reading the Activity Log: every one of those needs a real, signed-in administrator whether the switch is on or off. That list used to be the shared PIN's job. It is the one part of the model you cannot turn off, and it is deliberate — turning a switch off should relax a policy, never delete a gate. The rollout order is still **approve your machines** → **check they work** → **then turn it on** — on the **AV Central Settings page** → **Trusted machines** since Beta 40 (it was Admin → General → Trusted machines before that, and the server's start-up log still says so) — and turning it off again is one click and asks no questions.

› The three levels

LEVEL	WHAT IT CAN DO
Viewer	Sees everything — every page, every widget, every device, every report — and changes nothing, with one deliberate exception: a Viewer can comment on a cue , and resolve a comment, because "that camera cue is wrong" is worth hearing from whoever noticed it. This is what everybody is unless something says otherwise.
Editor	The whole Sunday-service surface . The Cue Sheet and every live cue operation (arm, fire, lock, the cursor, Calling Mode, the guest share link), stage messages, the incident log, acknowledging alerts, maintenance tickets, clip markers and headcounts, signage hold and paging, ticking off Run of Service, running the readiness check, and rearranging any dashboard or saving a layout preset. Plus four Admin tabs : Service Plan · PP ↔ PCO Links · Mic Assignments · Layout Presets.
Admin	Everything an Editor can do, plus all of Admin — Integrations, Devices, Alerts & Notifications, Roles & Positions, Run of Service structure, Screens & Displays, Files, Pre-Service Readiness, demo/live mode, software updates, settings import and export, trusted machines, and the Activity Log . An Admin also gets the Screens group inside the position picker in the top bar — one click to any of the seven link-only wall pages. Until Beta 39 that was a separate Screens ▼ button; the two controls are now one dropdown, positions above a dividing line and screens below it, and the Screens half is absent from the page entirely for anyone who is not an Admin.

i What an Editor actually sees on the Admin page. The page has **14** tabs. An Editor sees the four named above and nothing else — and on a **live** install, **three**, because **Service Plan** is the demo-plan editor and has hidden itself in live mode since Beta 3 for everybody. That is not a fault: on a live install an Editor opening Admin gets PP ↔ PCO Links, Mic Assignments and Layout Presets. Anything else they try is refused by the server, not merely hidden.

i Nobody is ever asked for a level they already hold. An Editor holds Editor for as long as that browser session lasts, so every cue operation, stage message, incident, ack and layout change is silent — no modal, no PIN, nothing to type while the service is running. An Editor is asked a question only when they reach for something that is genuinely Admin.

- i **Two places the split is finer than "four tabs". Run of Service:** Editors tick, Admin defines — checking off your own pre-service item is an Editor action, while adding, removing, renaming or reordering the checklist is Admin, and so is resetting it. **Push live:** an Editor's publish applies *only* their own four tabs and leaves anything an admin has pending still pending, so an Editor can never ship somebody's half-finished Integrations change on a Sunday. An Editor's copy of the settings also arrives with every credential stripped out of it — no PINs, no secrets, no integration passwords.

› What happened to the shared PIN

It was retired during the merge, and it is worth understanding why the order mattered. The PIN went through two demotions before it was deleted. First it stopped granting Admin and granted only Editor, so it could no longer open a settings door. Then the 43 actions that had always leaned on it were moved onto a real admin identity — and only once every one of them had somewhere stronger to stand did the PIN itself go. Deleting it in one step would have left the most consequential routes on the dashboard, restarting the server among them, answering any anonymous caller on the network.

What this means in practice. Nobody types anything to run a service. A volunteer at an approved booth machine holds Editor on arrival and is never prompted. Somebody who needs Admin signs in to AV Central as an administrator, or works from a machine approved at Admin level. And a person who cannot be raised to what they are reaching for is told so in a sentence — the dashboard no longer shows a box that cannot open the door, because typing a correct secret and being refused anyway reads as a broken dashboard rather than a permission boundary.

- ! **If a shared PIN is written on a card at FOH, it is now just a number.** It grants nothing. Take it down — a retired credential left lying about teaches people that credentials do not matter. The only PIN still in use anywhere in the dashboard is the **parent-paging PIN**, which guards the paging screen and its history and has always been a separate thing.

› Trusted machines — a machine that never has to sign in

AV Central → **Settings** → **Trusted machines** (it was Admin → General → Trusted machines until Beta 40 — the card moved, nothing else about it changed, and it still writes this dashboard's own access settings rather than AV Central's). A trusted machine is a browser that holds a level on every visit, with nothing to type. Each entry is **named**, holds **one level** (any of the three), and is **revocable on its own** without touching any other. It is meant for the machines that stay in the building — the booth Macs, the lobby laptop — not for phones.

- 1 Press **+ Add machine**, give it a name you will still recognise in six months ("FOH booth Mac", "Video booth", "Josh's laptop"), and pick its level.
- 2 On that row, press **Copy link**. The link is this dashboard's own address plus `/api/access/trust?k=<token>`. The card never shows the token unless you press **Reveal**, which puts it on screen for twenty seconds and then hides it again by itself.
- 3 Open that link **once**, in the browser on the machine that should hold the level. It lands you on a normal dashboard, and that browser holds the level from then on.

WHAT TO KNOW

WHY IT WORKS THAT WAY

The link is reusable, not single-use

Deliberate, and the opposite of the AV Central hand-off token in §6. A booth Mac whose browser data gets cleared has to be re-enrollable by whichever volunteer is standing in front of it. Needing an admin to mint a fresh link at that moment is exactly the Sunday-morning failure this feature exists to prevent, so opening the same link again simply works.

Changing a machine's level takes effect on its next request, with nothing to re-visit	The cookie carries only the machine's id — never the level — so the level is read live from the configuration every single time. Promote a machine from Editor to Admin and its very next page load is at the new level. Nobody re-opens a link and nobody is signed out.
Revoking drops that browser to Viewer on its next request	Revocation is removal: the cookie simply stops matching anything, so there is nothing to uninstall, no cookie to clear and nobody thrown out mid-service. The link stops working permanently — re-enrolling a machine later means adding it again and pasting a new link.
Treat a link like a key	Because it is one. It grants its level to any browser that opens it, for ten years. Don't paste one into a group chat, and don't email an Admin link to somebody who only needs Editor — make them an Editor machine instead.
Renaming or re-leveling does not re-issue the token	Rotating it on a rename would silently un-enrol the machine you were promoting. Revoke-and-re-add is the rotation path, and it is honest about what it does.

! Name the single point of failure, because it is a real one. With AV Central switched off, an Admin-level trusted machine is the **only** way to reach Admin. One such machine is one cleared browser — or one spilled coffee — away from nobody being able to reach the Admin page at all, and the way back from there is hand-editing `server/data/config.json`. Three things fix it, and all three are worth doing *before* the switch goes on. **One:** enrol **at least two** machines as Admin — the booth Mac and Josh's laptop. The card's own chip goes green at two and amber at one, so it tells you where you stand. **Two:** keep an enrolment link somewhere reachable **without** the dashboard — a password manager, **not** a bookmark in the browser on the machine it enrolls; that bookmark dies with the machine it was supposed to rescue. **Three:** the upgrade itself created one Admin-level machine called **Recovery** and printed its enrolment link into the server's start-up log, so this dashboard can never have come up with no path back to Admin at all.

i One honest limit on the Recovery link. It is printed **once** — on the upgrade that introduces the feature (or a fresh install's first boot) — and restarting the server later does not print it again. If it scrolled past unread it is still in `server/data/config.json` under `access.trustedMachines`, which is a recovery path but not a pleasant one at 8:15 on a Sunday. Copy it somewhere safe now, while nothing is wrong.

i Why a machine can out-rank the person standing at it. When more than one thing could decide your level — an AV Central account or a trusted machine — the **highest wins**. So somebody with view-only AV Central access, standing at the Admin-level booth Mac, is Admin. That is deliberate: a trusted machine is a property of the **room**, not the person, and anyone who can physically sit at the booth Mac can already edit the configuration file by hand. Capping the machine at the person's level would be theatre, and it would stop the booth Mac being the way back in.

› What it looks like with AV Central switched off

This is a **supported configuration**, not a half-finished one, and it is how the dashboard sits today. It works like this:

- **Everyone is a Viewer** by default — every screen in the building, every phone, every wall TV.
- **An Editor-level approved machine is an Editor**, which is the whole Sunday-service surface. No volunteer needs an account to run a service — they need to be sitting at a machine you have approved.

- **Admin is reachable only from an Admin-level trusted machine.** There is no other path — no link, no password, and no PIN, because there is no longer one to type. This is why the two-Admin-machines rule above is not advice.
- **No "Sign in with AV Central" button appears anywhere.** The dashboard only ever offers that door when sign-on is switched on and the secret has actually been pasted in, so with it off there is nothing to click and nothing to explain to anybody.

i **The trade, stated plainly:** with AV Central off you lose the per-person half of this entirely. Everybody at Editor is the **same anonymous Editor**, and the **Activity Log** (AV Central → Settings → Activity Log since Beta 40; see the Admin Training guide) can then only name the **machine** that acted, never the person at it. It says so in those words rather than guessing at a name. If you want "Josh pushed that live" rather than "the Booth Mac pushed that live", AV Central accounts are what buys it. That is the only thing you give up; nothing is degraded.

› Switching on AV Central sign-on, briefly

The full trust model, every refusal case and the security trade are in §6. In two lines: tick **Enabled** in **Admin → General → AV Central sign-on**, paste in the secret that was generated on the AV Central Mac in its own `server/data/hub.json` — carried across by hand, because AV Central never transmits it — and **Push live**. Once it is on, AV Central's own **Service** module access becomes the level: admin → Admin, edit → Editor, and anything else → Viewer.

! **One value on the AV Central side decides whether any of this lines up:** its `serviceDashboardUrl` must point at **this dashboard's real LAN address**, not `localhost`. That single field is what lets AV Central's position picker read this dashboard's live position list instead of falling back to a stale hard-coded five — three of which have never existed here. Get it wrong and everything else can be perfect while people are still assigned positions that don't exist and land on the wrong dashboard. Ask whoever manages AV Central accounts to set it, and to set it again if this dashboard ever moves.

› Every hidden page, in one list

Admin → Screens & Displays → Hidden pages. Seven pages have no entry in the nav on purpose — they are screens on walls, not places an operator browses to — and their addresses used to be scattered through the help text of four different cards. They are now one table with a copy button on every link, ready to paste into the browser on the machine that should show it. The links are absolute, because the whole point is pasting them somewhere else.

The same seven are also one click away in the top bar, for an Admin. Since **Beta 39** the old **Screens ▼** button is gone and its rows live inside the position picker, under their own **Screens** heading below a dividing line, each with its address beside it. Two groups, visually separated on purpose: a *position* and a *screen* are different kinds of thing and nobody reaching for "Lighting" should land on `/signage` by muscle memory. **The Screens group carries its own Admin check** and is absent from the page — not hidden by styling — for anyone below Admin. One consequence worth knowing rather than rediscovering: with access control switched **off** the server reports everyone as Admin, so on a switched-off install the group shows for everybody. That is a change from Beta 38, where it showed for nobody, and it is accepted because these seven pages are ungated reads anyone could already open by typing the address. The boundary is the server, not the menu.

PAGE	ADDRESS	VARIANTS
Stage display	<code>/stage</code>	<code>?tv=1</code>
Band board	<code>/board</code>	<code>?tv=1</code>

Signage	<code>/signage</code>	<code>?tv=1</code> · <code>?screen=<name></code> — type the name into the box on that row and copy the built address
Guest Services / Room Counts	<code>/paging</code>	—
Parent Paging	<code>/parents</code>	—
Calling Mode	<code>/calling</code>	—
Calling Mirror	<code>/calling/mirror</code>	—

! **Give every signage TV its own name — this card is where you do it, and here is why it matters.** A TV opened at plain `/signage` calls itself `default`, so **two unnamed TVs are clones**: they hold, resume and jump pages together, because every Companion button and every on-screen hold is aimed at the one name they share. Name each screen — `/signage?screen=lobby`, `/signage?screen=atrium` — and they move independently. Use plain lowercase names with no spaces, and use the **same** name in the `&screen=` part of the signage buttons under Admin → Alerts & Notifications → Stream Deck / Companion, or that button will look as though it did nothing.

› `?tv=1` is a display flag, and nothing to do with access

`?tv=1` makes a page a **full-bleed kiosk display**: no left rail, no top bar, and nothing that asks the room a question — no sign-in offers, no level chip, no permission chrome of any kind. That is all it does. It **never grants a write**. It is a query parameter anyone can type onto any address, including `/admin?tv=1`, and `/admin` is still fully gated when they do — the server ignores `?tv=1` completely when it decides what somebody may do.

i The same silence applies automatically, with no flag needed, to the seven link-only pages above and to a guest cue-sheet share link. A wall TV has nobody standing at it to answer a modal, and a modal it cannot dismiss is a dead screen in front of the room. Those pages are still able to do everything they need, because the only writes they can reach — parent paging and signage control — have their own credentials and were never part of this.

› The new Admin position

This release adds an eighth **position** called **Admin**, alongside the seven you already have. A position is a **widget layout** — which tiles you see and where — and this one is arranged around the whole-plant picture rather than one console: clock, pre-service countdown, alerts, device status, the device log, vendor status and maintenance tickets. It is hidden from the position picker for anyone who is not Admin, so a volunteer's dropdown does not grow a confusing eighth option.

! **Holding the Admin position grants nothing whatsoever.** Level and position are different axes, and this is the one place in the dashboard where both words are "admin". A **level** (Viewer / Editor / Admin) is what you may *do*. A **position** (Audio — FOH, Lighting, Admin) is what you *see*. Anyone can put themselves on the Admin position by hand and they still have exactly the level their account or their machine gives them — every route checks the real level, every time. The picker filter is tidiness, not a gate, and it should never be mistaken for one.

✓ **Lay it out how you want it.** The tiles it arrives with are a starting point chosen so the position isn't an empty grid on the first open — not a design. Switch to it, press the  button, and drag, resize, add and remove exactly as on any other position. Desktop and phone layouts are stored separately, as always.

8 Configure the room

› Roles and positions

Admin → **Roles and Positions** manages the list (Producer, Audio — FOH, Video/Streaming, Lighting, Stage Manager come built in; add your own, like Broadcast Director). Each role's widgets and layout are edited on the dashboard itself: click your position name in the dashboard header to switch to the role you want to edit, press the  button, then drag by the title bar, resize from the corner, add with "+ Add widget" (type to search the list), or remove with the **X**. Desktop and phone layouts are stored separately.

New in Beta 22 — an Admin position. The upgrade adds one more entry to this list, called **Admin**: a whole-plant layout (alerts, device status, the device log, vendor status, maintenance, clock and countdown) for the person who isn't sitting at a console. It arrives with tiles in it so it isn't an empty grid, and it is meant to be rearranged. It is hidden from the position picker for anyone below Admin level, and **holding it grants nothing** — a position is a layout, never a permission. See §7.

Standby warnings — new in Beta 19. Each position has a **Standby warnings** setting on this tab: *off*, *high-risk cues only* (the default), or *all cues*. It controls whether that position is told a cue of theirs is coming — the **Version History** document's Beta 19 entry says what it reads and when it stays quiet. The lead time is one number for everyone, in Admin → Alerts & Notifications → Thresholds.

› Mic assignments

Admin → **Mic Assignments** maps people to hardware with drag-and-drop: pack pool on the left, this week's Planning Center team (with photos) in the middle. Packs display the name physically programmed on the receiver (Wireless Workbench channel name) when the receiver is reachable. This drives the Mics page, the Wireless widget, battery alerts, and dLive fader labels.

Printing the sheet — moved in Beta 20. ↓ **Print assignment sheet** sits on this tab, next to the roster you have just built. It downloads a one-page US Letter PDF of who is on what. **It is no longer on the Mics page** — the button that used to be in that page's header has gone, because `/mics` is also the wall/TV view and a file download is dead weight on a screen nobody is standing at. The sheet is still an ordinary ungated download at `/api/mics/pdf`, so a bookmark at the rack works. It renders **what is live**, not this tab's unsaved draft — press **Push live** first, or you will print last week's roster.

› Devices to watch

Admin → **Devices**: name, IP, type, and room for everything worth monitoring — consoles, Dante switches, encoders, the lighting console, the stage TV. If it has an IP, list it. The  button on each row holds the full record: **switch/port it's plugged into, cable label, MAC address, and notes** — these also power the switch port maps on the Status page (mark switches with the "Port map" checkbox).

New in Beta 16 — Admin → **Devices** → **Device stability**: a read-only table of how many times each device has dropped off the network inside the alert window, sortable by clicking a column heading and defaulting to the worst offender first. This is where you go *after* a service to see whether something has been flapping all morning; the same count also shows as a chip beside the device's name on the Status page so it's catchable during one.

QR stickers for gear. Once a device is in this list you can print a sticker for it, stick it on the gear, and anyone who scans it can report that it is broken without signing in to anything — the sticker knows which device it is on, so nobody has to describe "the amp in the amp room, the middle one". Printing, placing and revoking them is a section of its own in the **Device Configuration Guide**.

› Alerts and incident buttons

SETTING	DEFAULT	WHAT HAPPENS
Low battery	30%	Warning toast + alert entry naming the pack and person
High SPL	95 dB sustained 10s	Warning so FOH sees it before the decibel-meter person in row three does
Device down	On · 60 seconds	Brief blips heal silently. Once a device has been down for the window you set, ONE critical alert fires with the auto-diagnosis built in — and it auto-acknowledges when the device is seen again.
Stream drop	On	Critical alert if a live platform goes dark mid-service
Device drops	5 drops / 60 min	New in Beta 16. A device that keeps dropping off the network and coming straight back never stays down long enough to trip the "Device down" row above, so it used to pass unnoticed. Cross this count inside the window and ONE Warning (not Critical — it never takes over the screen mid-service) fires. Both numbers are editable here; tune them rather than waiting for a release.
Slow device	250 ms sustained 120s	New in Beta 17. A device answering in 400 ms is "online" but not in any useful sense — on this network that usually means a failing link, a saturated uplink, or a dying NIC. Sustained breach raises ONE Warning and records a slow entry in that device's history; recovery re-arms it without a second alert. The sustain is held to a 40-second floor no matter what you type here, because devices are probed every 20 seconds and fewer than three breaching probes in a row isn't evidence of anything.

i **New in Beta 17 — grouped alerts.** When several devices go down together for the same reason, you now get **one** alert naming the shared cause and the devices behind it, instead of one per device all saying the same thing. Grouping is deliberately conservative: it only happens when the auto-diagnosis actually blames a shared switch or room, never on timing alone. Two unrelated devices dropping at the same moment still produce two alerts. If a switch goes down and you still get one alert per device, the devices' switch labels don't agree in **Admin** → **Devices** — that's a data fix, not a bug.

i The drop count is a **floor, not a total** — a drop shorter than one probe cycle never registers at all, so the real number may be higher. A whole network segment going down takes several devices with it, so expect several of these warnings at once when that happens.

The incident quick-log buttons ("Mic dropout", "Slide glitch", ...) are editable in the same tab. Logged incidents are time-stamped against the service and appear in that week's report.

i **Changed in Beta 17 — describing an incident.** Tapping **Other** now opens a description box under the buttons instead of logging a bare "Other" nobody can interpret later. Every other button still logs in one tap, and an empty description still logs — the timestamp is the part that can't be reconstructed afterwards. Every logged incident also gets a 📝 to add or correct its description later, which is the point: nobody types a sentence while the stream is stuttering. Edits made in Admin to this button list now appear immediately instead of waiting for a page reload.

› Preset stage messages — new in Beta 17

Admin → **Screens & Displays** → **Stage display — messages & notifications** → **One-tap messages** (this card was on the General tab until Beta 19): the three or four messages sent every week — "5 minutes", "Wrap up", "Stand by", "Go to black" ship as defaults — become one-tap buttons on the Stage Display Message widget. Edit the list here; free-typed messages keep working exactly as before. This is additive, so an operator who prefers to type is unaffected.

› Which graphics machine feeds each service item — new in Beta 17

Admin → **Links** no longer matches ProPresenter presentation *names* against Planning Center item titles. Instead each service item names the **GFX machine** whose live presentation should appear in that item's Cue Sheet slide block, and the Cue Sheet reads that machine directly — the same data the GFX widgets show.

! This needs one pass from you after upgrading. Your existing links were carried across as item rows with **every machine left blank on purpose** — a presentation name says nothing about which of the three GFX machines has it loaded, and guessing would silently attach an item to the wrong screen's deck. Until you choose, those items show no slide block, exactly as an unmatched link did before. Set them in **Admin** → **Links**, or from the Cue Sheet itself with the  **source** button on the item.

› Screens that should be checked in — new in Beta 17

Admin → **Screens & Displays** → **Screen check-in** lists every browser currently connected — which page each is showing, and when it was last heard from. Kiosk screens (the stage display, band board, paging screens, and anything opened with `?tv=1`) are recognised automatically; anything else is treated as somebody's phone or laptop. Use the **Expected** checkbox to mark any other screen that ought to be on every Sunday.

i This is what makes the pre-service **Stage & board displays** readiness check real. Before Beta 17 that check read a list nothing ever wrote, so it passed every single time — including with every screen in the building switched off. It now goes amber and **names** the missing screen. Expect it to be amber right after a restart until the screens come back up; that's the check working, and it never blocks a service.

! The same class of fault was found in the connector rows, and fixed in Beta 45 — so expect the board to be less green than it was. Every readiness row that tests a connector used to be marked **PASS** unless the connector *crashed*. Eight of the shipped connectors report a failure by returning an error instead of crashing, and for **Smaart** and the **ProPresenter stage** connector that is the only failure they ever produce — so those rows could not go red at all. The **AJA HELO** check was the worst case: it discarded its own errors and printed "Connected" with the encoder unplugged. The board now reads what the connector actually returned, and it uses the same verdict as the **Test connection** button on the Integrations tab, so the two surfaces can no longer disagree about the same connector. **Rows that have always been green may legitimately go amber or red on your first run. That is the correction, not a regression** — nothing changed in your rig on the day you updated. A check that cannot be read at all is now amber with the reason on the row, rather than a pass. A failing **critical** row takes the overall verdict to **NO-GO**, which is still advice: readiness has never blocked a service and does not now.

› Service reports

Admin → **General** → **Service reports**: reports archive only when Planning Center LIVE actually drove a service (so weekday testing never produces junk reports), and only when it ran longer than the minimum you set (default 15 minutes — false starts are skipped).

9 Updates, backups, and troubleshooting

› Software updates — drop a zip, done

AV Central → Settings → **Software update**: drop a release zip on the card and the platform installs it over itself, restarts into the new version, and keeps every setting (your configuration never lives inside the app files, and a settings snapshot is saved automatically before each update). The version that's running is always shown next to the Admin title.

! ⚠ **Moved in Beta 39 and Beta 40 — there is now exactly ONE update button and ONE restart button for the whole platform, and both are on the AV Central Settings page.** AV Central and the Service Dashboard are one application on this install, so the old pair on each side were two buttons driving the same single server — a trap rather than a convenience. The dashboard's copies have been retired and now answer with a sentence naming where the control went, rather than a bare "not found". **Restarting takes the whole platform down for a few seconds**, the Service Dashboard included, so never do it during a service. Nothing in `server/data` is touched and every screen reconnects on its own.

› Backups

Back up one folder: `server/data/` holds your entire configuration and every service report. Copy it to Drive or a NAS weekly (a scheduled robocopy/rsync of that folder is plenty). To reset the app to factory defaults, stop the server and delete that folder.

› The platform backup — and what it was missing until now

There is also a one-file backup you can download from Admin, and it covers the whole platform rather than one venue. It now contains the live settings and the unpublished draft, **every service's cue sheet** — the cues, highlights, notes and pasted script text — the saved cue templates and the whole cue library, how often each cue has been used, the service event log, the device incident history behind the ⚠ badges, the QR sticker tokens, the parent-paging history, the accounts, the shared equipment and fixture libraries, the organisation's name and logo, and every archived service report.

! **Read this if you have an old backup file saved anywhere.** Until Beta 38 that file contained the two settings files and nothing else. **No cue sheet, no cue template, no cue library, no event log and no incident history was in it** — the file whose whole purpose is to make "take a backup first" mean something did not contain a single cue. A backup taken before Beta 38 will restore your settings and will not restore a season of cue work. If a copy from before this release is the only one you have, **take a fresh one now** and treat the old one as settings-only.

Who can download it. Only the account that owns the whole system, signed in as a person. Not a dashboard admin, not an approved machine, and not "the access switch happens to be off" — because the file contains every password and sign-in the system holds, including the key that signs approved-machine cookies. Treat the downloaded file exactly like the password list it is; the archive says so about itself, in plain words, in a note inside it.

Two things it deliberately leaves out, so you are not surprised by their absence: guest cue-sheet share links, which expire after a day and would only ever restore as dead links that look live; and the diagnostic device log, which is about the machine you are replacing rather than the install you are restoring.

SYMPTOM	FIX
Page won't load from another computer	Confirm the host is running (<code>http://localhost:4400</code> on the host itself), then check the firewall allows inbound port 4400 on the host machine.

Widgets frozen, red dot in the top bar	The browser lost its live feed. It reconnects automatically; if not, refresh. Check the host didn't sleep — disable sleep on the host machine.
Test connection fails on a cloud API	Re-check the credential (most failures are a copied space). Confirm the host has internet — the Status page's vendor section is a quick tell.
ProPresenter connector fails	Network must be enabled in ProPresenter settings, and the dashboard host must be on a VLAN that can reach the ProPresenter Mac on the port shown in Settings → Network (1025 unless someone changed it).
A Cue Sheet item shows no slides	Since Beta 17 an item names a GFX machine , not a presentation name — no name matching is involved. Check the item's  source (or Admin → Links): no machine set means no slide block. If a machine is set and the item is still quiet, the block says why — machine unreachable, or a pinned presentation that isn't on that machine right now.
Battery/RF shows "—"	The receivers aren't answering the dashboard host, or the unit/channel numbers don't line up. Check the receiver IPs and unit names in the Shure connector against the unit column in Mic Assignments, and verify with WWB that the receiver answers on the network. The Wireless widget names any receiver that failed to connect on a "not reachable — check TCP 2202" chip.
The printed sheet says "Mic 7" instead of the name on the pack	The sheet prints whatever Channel Name the receiver reports, falling back to the configuration label when there isn't one. Two possible causes, and the Mics page tells them apart: battery and RF present → the receivers are answering and the Channel Names simply aren't programmed (set them in Wireless Workbench); battery and RF blank too → the receivers aren't reachable at all, so fix the network path first.
Wrong port or 4400 is taken	Start with a different port: <code>PORT=8080 npm start</code> (macOS/Linux) or set PORT in the Task Scheduler action's environment.

10 Quick reference

Tape this to the FOH desk

DASHBOARD

http://10.0.5.20:4400

ADMIN

.../admin + Admin level

UPDATE

Admin → General → drop the zip

STAGE DISPLAY

.../stage

FIRST START

node server/index.js

BACKUP FOLDER

server/data/

TV MODE

add ?tv=1

AUTO-RESTART START

npm start

DURING THE SERVICE

WHERE

Log an issue the moment it happens

Incident Log widget — one tap

Message the stage

Stage Display Message widget → Send to stage

Acknowledge an alert

Alerts widget → Ack (device alerts ack themselves when the device recovers)

After service: pull the report

Reports page — it generates automatically when PCO LIVE ends

This document references dashboard version 1.1.0 Beta 52. What changed in each release lives in the separate **Version History** document.