Myro:Home Helper

Getting Started

Version 5.50

Connect your controller, complete the first synchronization, and choose the services that make sense for your home.

Myro Control, LLC | myrocontrol.com

Contents

Welcome to Myro:Home Helper

- Before You Begin

Connect Your Controller

- Control4 controllers
- Complete the First Synchronization
- Make the Connection Persistent

Choose What to Enable

- Room Photos from Helper
- HomeKit in Brief
- Siri and Shortcuts in Brief
- Logging and Graphs in Brief

Browser Control and History

Wall and Countertop Tablets

- How the Tablet Layout Turns On
- The Wall Tablet Address
- Keep the Screen Awake
- Add to Home Screen
- Cameras on a Tablet
- Signing In a Tablet
- Supported Tablets

Apple TV

- Pair an Apple TV
- How Long a Pairing Lasts
- See and Revoke Paired Apple TVs
- Arming and Disarming from the Apple TV
- If Something Does Not Work

Enable Push Notifications

Everyday Operation

- Reaching Helper Away From Home
- Remote Access with Tailscale
- Savant Outside the House
- Connecting at Home with an Outside Address
- When Something Does Not Look Right
- Learn More

Welcome to Myro:Home Helper

Myro:Home Helper 5.50 is an always-on macOS companion for HAI / Leviton Omni and Lumina controllers, NESS / ELK panels (preview), Savant systems (preview), and Control4 systems through Control4 Link (preview). It maintains local connections and adds automation and configured integrations. Features follow the controller family; Savant and Control4 use rooms and media activities rather than panel objects.

At its simplest, Helper connects to your controller and runs quietly on a Mac. You can then add only the capabilities you want:

- Apple Home and Siri through the built-in HomeKit bridge
- Mac Shortcuts and Siri: run shortcuts from scripts, and control lights, macros, status and your own script actions from the Shortcuts app
- Myro:Home and Myro:Home Pro mobile access and push notifications
- Historical temperature, humidity, zone, and Ethernet-quality logging
- Local graphs in a web browser
- JavaScript automation and schedules
- Lutron, Sonos, MQTT, Vestaboard, LaMetric, weather, and energy integrations
- A local HTTP API and MCP access for supported AI clients and automation tools

> Helper complements the controller; it is not a replacement for the panel's programming tools. Existing panel automation, names, links, and device configuration remain in the controller.

What's New in 5.50

- **Myro:Home Pro for Apple TV 2.0.** Helper 5.50 adds pairing by code, so an Apple TV can show your rooms, cameras, favorites and a Today row of recent activity from the last day, without a sign-in. Allow the request on the Mac, and revoke any paired Apple TV from **API > Control Login > Paired Devices...**. See **Apple TV** below.
- **Shared favorites in the browser.** Star something once in browser control and every browser and device signed in to that controller shows it. See **Browser Control and History**.
- **Right now.** The browser's **Overview** starts with a short row of the things your household usually does at about this time, built from the taps Myro:Home Pro and the Apple TV share with Helper. One tap does it; nothing ever happens on its own. **Clear usage history** in Settings forgets it all. See **Right now** under **Browser Control and History**.
- **Add UniFi Protect cameras from the browser.** In **Cameras**, choose **Add from UniFi Protect**, sign in to your Protect console, tick the cameras you want, and they are added.
- **Camera events from UniFi Protect.** Stay signed in to your Protect console and the Timeline shows when a camera sees a person (or a vehicle, an animal, a package or motion, if you choose) and when someone rings a Protect doorbell, with Protect's own picture if you allow it. You choose what each camera reports; Myro:Home Pro and the Apple TV show the same events. See **Camera events from UniFi Protect** under **Timeline**.
- **Timeline in the browser.** A new **Timeline** page, right after **Overview**, shows the last 8 or 24 hours at home: a one-line summary, moments such as arrived home, left home, doorbell, movie night and good night, five lanes (security, doors and motion, lights, climate, audio and media), and a draggable playhead

that rewinds the room cards to show what was on at that minute. It adds gentle notes (a light left on, a door left open, a room well off its setpoint, a door opened while armed), marks offline gaps, and keeps a camera picture for 24 hours for each doorbell ring when a camera is in that room, with a switch to turn that off. For HAI / Leviton, NESS / ELK, Savant and Control4. See **Timeline** under **Browser Control and History**.

- **Wall and countertop tablets.** In landscape, a touch tablet shows browser control laid out like Myro:Home Pro on iPad, with a rail of views and two panes; **Settings > Layout** or `?layout=` overrides it. The `/helper/tablet` address adds ambient dimming, an overnight reload and an always-kept sign-in for a tablet on the wall. See **Wall and Countertop Tablets**.
- **Timeline from scripts.** Your scripts can add notes and named moments to the Timeline, quiet a chatty sensor, or take it over entirely, and the Timeline now catches lights the panel changes quietly, shown with a "~" time. See **Scripts and the assistant** under **Timeline**.
- **Doorbells.** Mark any zone, flag or output with **Treat as doorbell**. Each press appears in the Timeline and the Apple TV's Today row.
- **Arm and disarm from Apple TV.** A paired Apple TV can arm and disarm with the panel code when the Control Login allows security control. It cannot bypass zones or force arm. See **Arming and Disarming from the Apple TV**.
- **Savant homes in the browser and on Apple TV.** A Helper that runs only a Savant system now lists it for sign-in, with **Run** and **Stop** for its activities, and you can pair an Apple TV with it.
- **Control4 Link (preview).** Add a Control4 system alongside your other controllers: rooms and Watch/Listen activities with live feedback, Apple Home activity switches, scripting, browser sign-in with **Run** and **Stop** for its activities, the Timeline, Apple TV pairing, and a new **Control4 Controller Guide** in the Help menu. See **Control4 controllers** below.
- **Pair Myro:Home Pro with Savant and Control4 controllers.** Their **Connection** tab now ends with **Pair with Myro:Home Pro**, the same QR and typed code an ELK controller has. Pro then finds this Helper and shows its room photos for the controller (and, for Savant, its Timeline). The code is for Helper features only; Pro never connects to the Savant or Control4 system with it.

Added in 5.20

- **Live camera wall and shared setup in build 3040.** Visible camera cards play live video, with Cameras directly below Events. Status responses that reorder JSON fields keep the existing cards and streams. Playback recovers into available video after a buffer gap. Manage stored layouts in **API > Rooms & Cameras** and combine byte-identical photos in **API > Photos**.

- **Optional HAI receivers and Savant Custom Names.** HAI audio zones can opt into separately paired receiver controls through **File > HAI Audio Receivers (Prototype)...**, alongside existing HAP switches. Build 3033 fixes system-specific mute, including Russound, and keeps the opt-in light-style Home volume slider without an extra Mute switch. Use the slider's on/off for mute and Control Center's Remote for iPhone volume buttons. Savant HAP activities now offer a **Custom Name** for Apple Home and Siri. See the HAI and Savant controller guides for setup and limitations.

- **Savant reliability in build 3029.** The native Rooms pane identifies sources by type and variant and adds Play, Pause, Stop, Next and Previous for the selected active source. Buttons follow discovered commands; Refresh retries missing metadata. Room Off stops all confirmed active sources. Inventory refreshes on reconnect and every five minutes; unknown state remains unavailable. HomeKit and API status distinguish supported, configured and running features.

- **Savant controllers (preview).** Discover and trust a host, sign in with a Savant Local User, and control rooms and media activities with live feedback. No cloud token or agent ID is needed. Scripts can combine permitted controllers and configured integrations such as Vestaboard. See the Savant Controller Guide.
- **Shared photo library.** Manage reusable full-resolution photos in Helper and download copies through PRO's From Helper photo selector for supported HAI/ELK connections.
- **Updated Help and AI references.** Separate HAI / Leviton, NESS / ELK, and Savant controller guides explain each family's capabilities.

Included in 5.20

- **Siri and Shortcuts on the Mac.** Scripts can run any Mac shortcut and use its text result (`MyroSystem.runShortcut`). The Shortcuts app gets six Helper actions: **Set Light / Unit**, **Press Button / Macro**, **Get Object Status**, **Arm Security**, **Disarm Security** and **Run Script Handler**. Name a shortcut to run it with Siri. They run in the background, even while the Mac is locked, and a script's actions appear only after you approve that script in the script editor. See **Siri and Shortcuts in Brief** below and the Scripting Guide.

Controller support

- **NESS / ELK panels (preview).** Add an M1 / M1 Gold, EZ8 or EZ24 panel with a compatible Ethernet interface. ELK panels get Helper's push notifications and Myro:Home Pro Live Activities (arming countdown, alarm, tripped zones and panel troubles), the HomeKit bridge, scripts, the HTTP API, MCP and web control. See the separate NESS / ELK Controller Guide.
- Controller setup and supported panel features now have separate guides for each family.
- Everything Helper does for HAI and Leviton works as before.

Mac requirements and speech

- Helper now requires macOS 12 Monterey or later, and is built with Xcode 27 for macOS 27.
- Announcements use the Mac's System voice (System Settings ▸ Accessibility ▸ Spoken Content ▸ System voice). Choose a Siri, Premium or Enhanced voice there for the most natural sound; Helper follows a change on the next announcement.
- "Myro" is pronounced *Mai-row* (MY-row) everywhere Helper speaks.
- Scripts, the HTTP API and Sonos announcements can name a different voice for a single announcement.

Before You Begin

Use a Mac running macOS 12 Monterey or later on a trusted local network. Keep it powered on, logged in and awake for Helper services. Internet access is needed only for online integrations such as push notifications or weather.

Connect Your Controller

Choose your controller guide for hardware requirements, network settings and credentials:

Controller	Connection and feature reference
HAI / Leviton	HAI and Leviton Controller Guide
NESS / ELK	NESS / ELK Controller Guide
Savant	Savant Controller Guide - Local User sign-in, rooms, live status, and scripting
Control4	Control4 Controller Guide - Control4 Link setup, room activities, Apple Home switches, and scripting

For an ELK, Savant or Control4 controller, finish by pairing your phone. ELK shows the code under **Connection > Pairing**; Savant and Control4 show it at the bottom of their **Connection** tab under **Pair with Myro:Home Pro**. In Myro:Home Pro choose **Pair with Helper** and scan the QR code (or type the code) so the app finds this Helper. An ELK controller also receives its home address this way. For Savant and Control4 the code unlocks Helper features only (room photos, finding this Helper and, for Savant, the Timeline); Pro never connects to the Savant or Control4 system with it.

Select the controller's Connection tab, follow its guide, and choose Connect. Initial setup or **Sync Controller Data / Sync Now** downloads names and configuration into Core Data. Subsequent app opens and reconnects refresh lightweight status/values while retaining the saved inventory. Run a full sync after changing panel names or configuration.

For Savant, follow its controller guide: **Connection** provides host discovery, trust, and Local User sign-in; **Rooms** provides native inventory and controls. **Remember Local User in Helper Keychain** saves a successfully authenticated login, including when checked while connected. Auto-connect is a separate setting. The **HAP** tab lets you select Savant activities as Apple Home switches. The native Rooms pane provides full Savant media controls. Synced activities in the HAI/ELK browser Rooms view can also start/stop the matching configured Savant activity; panel commands remain separate. **Pair with Myro:Home Pro**, at the bottom of **Connection**, shows the code Pro scans to use this Helper's Timeline and room photos for the Savant controller; Pro's own Savant connection is unchanged.

Control4 controllers

Installer preview for Helper 5.50. While this release is pending, use only an explicitly supplied compatible Control4-enabled preview/TestFlight build for testing. Actual Control4 hardware acceptance is still pending. The driver declares OS 4.3.0 as its minimum, with CORE 5 / OS 4.3.0 / Halo as the initial target; other controllers are not claimed as tested. Commissioning needs owner-authorized compatible Windows Composer Pro and a project backup; normal operation needs an awake Mac running Helper on the home LAN. See the **Preview status and requirements** and **Frequently asked questions** sections of the Control4 Controller Guide for workstation tools, local TLS and installer setup.

Control4 Link connects Helper to a Myro connector that your Control4 installer loads into the project. Ask your installer to back up the project, install the Myro Control4 Link driver, and give you the private `helper-connection.json` file for your installation. Keep that file private: it contains a secret.

1. Click **+** beneath Helper's controller list and choose **Control4 Link**.
2. In **Connection**, choose **Import Trusted Connection...** and select the `helper-connection.json` file. Check the host, installation and certificate fingerprint with your installer. The connection uses pinned TLS and a credential unique to your installation.
3. Choose **Connect**, and turn on **Connect automatically when Helper starts** if Helper should reconnect after a restart. The **Rooms** tab lists your rooms and their Watch/Listen activities with live feedback. **Start Activity**, **Stop Selected Activity** and **Whole Room Off** control the selected room; Play, Pause, Stop, Next and Previous appear when the connector reports them.
4. To use Apple Home and Siri, open the **HAP** tab, choose the activities to expose (each with an optional **Custom Name**), check **Expose selected Control4 activities to HomeKit**, choose **Apply**, then pair in Apple Home with the displayed code. Activities appear as switches on a bridge named **MyroHome Control4** followed by your controller's name; one that is unavailable shows **No Response**.
5. To let Myro:Home Pro use this Helper's room photos for the controller and find the Helper on your network, scroll to **Pair with Myro:Home Pro** at the bottom of **Connection** and, in Pro's **Pair with Helper**, scan the QR code or type the code. The code is for Helper features only; Pro does not connect to the Control4 system with it.

A Control4 controller has rooms and activities rather than panel objects, so there is no security, zone or Sync Now step. In the browser it offers the same views as a Savant home, with the Timeline and **Run** and **Stop** for its activities (see **Control4 homes** under **Browser Control and History**), and an Apple TV can pair with it. Siri controls the exposed activity switches on and off; playback, volume and navigation are not exposed through those switches. Scripts use `MyroControl4`. Your Control4 project remains responsible for equipment control and routing, and Control4 app accounts and Connect subscriptions are separate from this connector. See the Control4 Controller Guide.

Complete the First Synchronization

For HAI / Leviton and NESS / ELK, after the first connection select **Sync Now** in the toolbar. Helper downloads the controller's names and object information for zones, units, buttons, thermostats, audio, and other supported objects.

Keep Helper open until the synchronization progress finishes. Large systems can take several minutes. The downloaded names are what Helper uses in Edit Objects, notifications, graphs, scripts, the API, and mobile discovery.

If an object is missing or you later rename something in the panel, connect to the controller and run **Sync Now** again.

Make the Connection Persistent

For normal panel operation, enable:

- **Automatically connect to controller** in the Connection tab
- **Start Myro:Home Helper automatically when logging into this Mac** at the bottom of the main window

The Mac must remain awake, logged in, and connected to the network for Helper-provided services to remain available.

With the login setting on, Helper also restarts itself if it ever stops unexpectedly. A small companion process installed by that setting runs alongside Helper and watches it. When Helper quits without being asked — a crash, or a Force Quit — the companion starts it again within a few seconds, so the controller connection, the HTTP API, and HomeKit come back without anyone going to the Mac. Quitting Helper normally, logging out, and shutting down are recognised as deliberate and are never undone, and a normal quit stops the companion as well, so nothing of Helper is left running. Opening Helper again starts the companion again. To stop Helper, quit it rather than force-quitting it.

If Helper fails repeatedly, restarts stop after five attempts in an hour and the delay between them grows. A fault that a restart cannot fix — a full disk, a damaged file — is then left alone rather than being retried forever, and the companion stops too; opening Helper again starts it afresh. Turning the login setting off stops the companion immediately.

Choose What to Enable

You do not need to configure every tab. Start with the controller connection, then add services as needed.

Capability	Where to Start	Purpose
Apple Home and Siri	**HAP** tab, then **Edit Objects**	Expose selected controller objects as HomeKit accessories
Mobile notifications	**Notifications** tab	Send selected controller events to Myro:Home apps
Historical data	**Logging** tab and **Edit Objects**	Record selected events and sensor values in managed CSV archives
Graphs	**Graphs** in the toolbar	View enabled sensor history and Ethernet quality in a browser
Automation	**Scripting** in the toolbar	Create JavaScript event handlers, schedules, and cross-system actions
Local integrations	**Add-ons** and integration windows	Connect Lutron, Sonos, MQTT, displays, weather, and energy services
API and MCP	**API > API Keys**	Create a controller-specific API key for trusted local clients
Browser sign-in	**API > Control Login**	Set a username and password for this controller
Timeline and doorbells	**Timeline** in browser control	See the last 8 or 24 hours at home; mark doorbells with **Treat as doorbell**
Apple TV	Myro:Home Pro for Apple TV, then **API > Control Login > Paired Devices...**	Pair an Apple TV with a six-digit code; see or revoke paired Apple TVs
Shared photo library	**API > Photos**	Manage reusable room and home photos for PRO
Shared room/camera records	**API > Rooms & Cameras**	Inspect, rename, hide or remove stored shared records; appears when records exist

Room Photos from Helper

For a supported HAI / Leviton or NESS / ELK connection, select the intended controller in Helper's sidebar, then open **API > Photos**. Choose **Import Photos...** to add one or more room or home images. Select a row to rename, replace, or remove it. Each controller has its own library; the heading names the library you are managing. Removing that controller also removes its Helper library.

New uploads of byte-identical images reuse the existing library photo even under a different name. To clean up older exact copies, choose **Combine Exact Duplicates...**, review the listed images, then confirm. Helper keeps one displayed entry per identical image and shares its stored bytes; existing photo IDs, room assignments and Savant source records remain valid. Different images are never merged by appearance or name. Explicit replacements keep the selected photo ID.

In PRO's photo selector, choose **From Helper**, then select a photo to download and save on that device. You can still choose the device library, camera, or clipboard. Downloaded photos remain available when Helper is offline. Changing or removing a photo in Helper does not change copies already saved in PRO; select it again to download the new version.

To add a photo to Helper from PRO, choose a new image from the device library, camera, or clipboard. On **Adjust Image**, turn on **Save a copy to Helper** before confirming the crop. This option appears when a paired Helper is available for the current controller. The photo saves on the device first, then a **Save to Helper** sheet lets you name it and choose the Helper if more than one serves that controller. Tap **Save to Helper** to add it to that controller's library. Closing the sheet or a failed upload leaves your device photo in place. If Helper asks for an update, install a build that supports photo uploads. A photo already selected from Helper does not offer to upload a second copy back.

Helper stores full-resolution JPEG masters with rotation applied and location/camera details removed. Imports are limited to 40 MB, 60 megapixels, and 16,384 pixels per side. Use an updated Helper and connect PRO to the matching controller to see its library.

HomeKit in Brief

For Savant, open **HAP**, choose the activities to expose, check **Expose selected Savant activities to HomeKit**, and click **Apply**. Pair using the displayed QR code or pairing code. ON starts an activity; OFF stops only that activity. Values follow Savant feedback, and disconnected or unknown activities show **No Response**. A unique activity match after a host upload preserves its HomeKit identity; missing or ambiguous matches stay unavailable.

For Control4, open **HAP**, choose the activities to expose, check **Expose selected Control4 activities to HomeKit**, and click **Apply**. Pair using the displayed code. ON starts the activity; OFF stops only that activity when Helper can verify the exact source, and a room or activity that is unavailable shows **No Response**. If the same activity is already exposed through another HomeKit bridge, choose one path to avoid duplicates.

The object steps below apply to supported HAI / Leviton and NESS / ELK objects.

1. Open the **HAP** tab and select **Edit Objects**.
2. Enable HAP only for the objects you want in Apple Home and assign the correct HAP type.
3. Enable the HomeKit Accessory Protocol bridge and start the HAP server.
4. In Apple's Home app, add the Helper bridge using the displayed setup code.

Restart the HAP server after changing which objects are exposed. Keeping the accessory list focused also makes the Home app easier to manage.

Siri and Shortcuts in Brief

Helper works with the Mac Shortcuts app in both directions (Siri and Shortcuts actions need macOS 13 or later):

- **From Shortcuts into Helper.** Add a **Myro:Home Helper** action to a shortcut: set a light, press a button or macro, read a zone, unit, thermostat, flag or output, arm or disarm an area, or run a script action you made with `ScriptEvents.onShortcut`. A script's actions appear after you turn on **Allow this script's handlers in Siri & Shortcuts** in the script editor; editing the script asks again. Actions run in the

background without opening Helper, even while the Mac is locked, so an always-on Helper Mac needs no one at it. The first time a shortcut uses Helper, macOS asks you to allow it.

- **Arming and disarming. Arm Security** needs no code, like HomeKit. **Disarm Security** needs your security code, which the controller checks; repeated wrong codes lock out further tries for a while. To disarm unattended, save the code in the shortcut, and keep in mind that anyone who can open that shortcut can read it.
- **From Helper into Shortcuts.** A script can run a shortcut with `MyroSystem.runShortcut` and use its text result. Start from **Templates > Utilities > Run a Shortcut**. The first time, macOS asks to let Helper control **Shortcuts Events**; you can change it later in **System Settings > Privacy & Security > Automation**.

Shortcuts on an iPhone can't use the Mac's actions; use the Myro:Home apps or Helper's HTTP API instead.

Logging and Graphs in Brief

Use the Logging tab to choose event and object categories, then use **Edit Objects** to enable logging and graphing for individual objects. Diagnostic logs and non-graph CSVs use bounded rotation. Sensor history follows a different policy: it is split into compressed archives, and all history is retained by default. Back up the CSVs together with their adjacent `.history` folders; keep-all storage will continue to grow over time.

Graphs are served locally from the Helper Mac. Open them with the toolbar button or from a Myro:Home Pro controller that has discovered this Helper.

Browser Control and History

Open `http://HELPER_HOST:9393/` using the IP address or hostname of the **Mac running Helper**, not the controller. Select the controller, or open **Control** from Helper's toolbar. Bonjour web links also open Control for their controller.

In Helper, select your controller, then **API > Control Login**. Enter a username and a password of at least 15 characters, choose **Save Login**, then type the password again when asked. Once a login is saved, the password shows as dots. To change only the username or the security setting, leave the dots as they are; type a new password only when you want to change it. **Open Control** opens the browser control for the selected controller. Use these credentials on the browser's **Control Login** sign-in. Each controller has its own login. Saving a replacement password or removing the login signs out existing login sessions. Helper stores a salted password verifier in this Mac's Keychain, not the password itself.

Existing API keys still work: choose **API key** on the sign-in page. Create or view keys under **API > API Keys**. A Control Login does not grant API or script-management access.

The browser provides an overview, a Timeline of the last 8 or 24 hours, favorites, room filtering, security, lights and outputs, climate, audio, messages and panel events. It uses the same typed commands as the HTTP API. Supported controls hold their requested value with **Updating…** while waiting for reported state. A rejected command reverts immediately; an accepted change that is not confirmed within 15 seconds returns to the last reported state with a warning. This is not proof the physical command failed. Security and lock status are never optimistically marked as changed. Disconnected or stale browser status disables controls.

Right now

Overview begins with a **Right now** row (the tablet layout leads with it): up to six things your household usually does at about this time on a day like today, each a one-tap shortcut like Recents. It is built only from what people did in the apps: the taps Myro:Home Pro 3.10 and Myro:Home Pro for Apple TV 2.0 send to Helper (Myro:Home and Myro:Home for Android follow). Schedules, wall switches, Apple Home and what the house did on its own never count. Helper keeps those taps on this Mac for 90 days, with no cloud, and scores them by the half-hour, weekday or weekend, whether the phone was at home, and how recent they are; each suggestion says why it is there, such as *usually around now* or *most weekday evenings*, and something that is already on, armed or locked drops down the list. A suggestion never acts by itself, and an arming, disarming or unlock suggestion still asks for your code. Browsers and wall tablets see the row but never add to it. **Show Right now on the Overview** in the browser's **Settings** turns the row off for that browser, and a tile's small **x** (**Not now**) hides that one thing for the rest of the day. Each app has a **Share usage with Helper** switch, on once it is paired with Helper; turning it off removes that device's taps from Helper. **Clear usage history** in the browser's **Settings** (the gear) shows how many taps each device has shared and forgets them all for this controller. Removing a controller from Helper removes its usage too.

Rooms, cameras and shared changes

API > Photos > Open Photo Manager… (also **Window > Photo Manager…**) opens a separate, resizable photo library window for the selected controller. Import, rename, replace, remove and combine exact duplicates there; the Photos tab and manager stay in sync. Each controller keeps its own window, and opening it again brings that window forward.

On the Helper Mac, **API > Rooms & Cameras** lists the selected controller's shared rooms and cameras: names, assigned-control counts, photo presence and camera hosts, without exposing camera credentials. **Open Editor...** (also **Window > Rooms & Cameras...**) opens a resizable native editor: pick a room to change its name, web visibility, photo (library, upload or default) and its controls from a searchable list with assign and show-in-web checkboxes; pick a camera to change every connection field with a snapshot preview. **Add Camera** also offers **Add from UniFi Protect...**: the same console sign-in and camera list as the browser view, so you can tick the cameras to add (and keep the sign-in for camera events) without leaving the Mac app. A UniFi Protect camera's editor carries the same **Camera events** choices as the browser's (Person, Vehicle, Animal, Package, Motion, Ring, Keep pictures and Same doorbell as), saved with the camera, and **UniFi Protect...** in the sidebar is the browser's UniFi Protect card: the kept consoles, **Disconnect**, and **Connect a Console...** for camera events. Panel rooms can be hidden or shown; rooms added here and cameras can be removed after confirmation (**Remove Room...** and **Remove Camera...** at the bottom of the editor; a panel room offers **Hide Room** instead). The pane's Rename and Remove shortcuts and **Open Web Editor** remain. The sidebar keeps both Add buttons visible, opens the first available item and preserves your selection during updates. A simultaneous edit is rejected and the editor shows "changed elsewhere" instead of overwriting. Devices still require their own reviewed sync before accepting layout changes.

On a HAI/ELK browser connection, open **Rooms** and select the **Edit rooms** sliders icon, then **Add room**, even when the list is empty. The edit toggle reveals Add room, Edit room and photo actions; choose the checkmark to finish editing. Room tiles highlight as a whole when hovered or focused. Clear **Show room in Helper** in its editor to hide a room outside edit mode; hidden rooms remain available there with a label. Name the room and assign its controls or cameras; **Edit room** changes an existing room. Removing a room keeps its controls and archived photos. **Change room photo** offers the Helper library, image upload, or the default photo. A room without a photo shows a generated background and an icon chosen from its name (a bed for Master Bedroom, a car for Garage, the first letter when nothing matches), the same look Myro:Home Pro and Pro for Apple TV give that room. Custom room and controller images are also kept under **API > Photos**.

Open **Cameras**, directly below Events in the sidebar, for a live wall: every card you can see plays live video (up to six at a time in one browser; the rest show snapshots that refresh about every ten seconds and keep their last frame). Use its **Edit cameras** sliders icon to reveal Add camera and each card’s Edit action; choose the checkmark to finish editing. Clear **Show camera in Helper** in its editor to hide a camera outside edit mode. These Helper visibility choices are shared across its browsers, preserved by PRO/Classic sync, and do not hide the item in the native apps. Each card carries a status pill: **LIVE**, Connecting, Reconnecting, Stalled, Busy (Helper is streaming to its maximum number of viewers), Offline or Unsupported. Nothing needs pressing when a camera drops: the player keeps retrying with increasing pauses, reconnects as soon as your network or the page comes back, and notices a frozen picture by itself. Select a card or **Watch live** to open it larger without reconnecting; the viewer adds **Unmute** (only when the stream has sound), a format menu (Auto, H.264, MJPEG), **Fullscreen**, **Reconnect** and a line showing format, size, how far behind live you are and bitrate. If your browser cannot play a camera's native format the player switches to H.264 or MJPEG on its own and remembers that choice for a day. Closing the viewer hands the stream back to its card. Leaving all camera surfaces stops every stream; room-assigned cameras keep playing in that room. Status updates and moving a player do not restart its stream. The PRO Myro logo animation appears while loading, pauses offscreen/in the background, and respects reduced-motion settings. Offline/Busy explanations stay visible during background retries. Brief visibility changes do not churn decoders, and only sustained recovery resets retry backoff. Helper pulls each camera once and shares it between all viewers and snapshots; audio is converted only while someone has unmuted it. Room-assigned cameras use the same cards. Helper can seed cameras from Extended Setup and share camera settings and room layouts with linked PRO and Classic apps.

Shared sync appears only when that controller has an available paired Helper. Without Helper, the apps retain their local room/camera behavior. Camera playback support depends on the source and configured stream format.

Add UniFi Protect cameras. In **Cameras**, choose **Edit cameras**, then **Add camera**, then **Add from UniFi Protect**. Enter your console's server address (including `https://`; `http://` works for adding cameras but cannot stay signed in), the username and password of a local user (or of your UniFi account, if it has no two-step verification), and choose **Connect**. In UniFi Protect, make sure RTSP is turned on for the cameras. Helper lists the console's connected cameras; tick the ones you want and add them. Cameras already in Helper are marked **Already in Helper**, however they were added (here, in PRO or by hand), and a camera without RTSP says to turn it on. Unless you tick **Keep signed in for camera events**, Helper uses the sign-in for that request only and does not store it; see **Camera events from UniFi Protect** below. A camera added this way needs no camera password. If PRO already shows your Protect cameras, sync them to Helper from PRO first (**Edit > Sync with Helper**) instead of adding them again here, so each camera exists once. Choose **Enter a camera manually** to type a camera's address instead.

Doorbells and history. On a zone, flag or output card, turn on **Treat as doorbell**. Helper then records each press (a zone becoming not ready or tripped, a flag going from 0 to any value, or an output turning on) and keeps three days of home activity per controller: doorbell presses, lights going on and off, security, door and motion events from the panel's event log, and the UniFi Protect camera events you choose. The Apple TV's Today row shows recent activity from the last day, and the **Timeline** page below shows the last 24 hours in the browser. Sensor history stays under **Graphs**.

Timeline

Timeline sits right after **Overview** in the browser's sidebar. It shows what happened at home as one picture: a one-line summary at the top (for example, *Evening at home. Home at 6:31 PM. Doorbell rang at 7:42 PM.*), then a bar for the whole period with the moments, lanes and playhead described below. It is available for HAI / Leviton, NESS / ELK, Savant and Control4 controllers.

8 h or 24 h. Choose **8 h** or **24 h** above the bar to see the last 8 or the last 24 hours. Helper keeps 24 hours of detail, so that is as far back as the timeline goes. Nothing is recorded before Helper first runs, so a new setup fills in over the first day.

Moments. Bursts of activity are grouped into a few named moments above the bar: *Arrived home*, *Left home*, *Good night*, *Good morning*, *Doorbell*, *Alarm* and *Movie night*, a panel scene by its own name, plus a busy room when a lot happens at once, and a person seen by a UniFi Protect camera (see **Camera events from UniFi Protect** below). Select one to see what it was made of: the lights, doors, arming and media events inside it, with the time of each. A moment keeps its name as it grows, so a long evening stays one moment rather than many.

Five lanes. Under the moments, each lane is a row of colored blocks across time:

Lane	What it shows
Security	Arming, disarming and alarms, by area
Doors & motion	Doors opening and closing, and motion in the home
Lights	How many lights were on at each point; taller and brighter blocks mean more lights
Climate	When the heating or cooling was running

Lane	What it shows
Audio & media	Music, audio zones and watched sources (Savant and Control4 activities, for example)

A lane appears only when the controller has that kind of device, so a Savant or Control4 home shows media and busy rooms but no doors or security. Hover over a block (or select a moment on a touch screen) for the details. Grey bands across the lanes mark **Offline** gaps, when Helper was not running or the controller was disconnected, so an empty stretch is never mistaken for a quiet one.

The playhead. Drag the **Now** handle along the bar, or select an event in a moment, and the room cards below the bar change to show the house at that minute: which lights were on, the temperature, what was playing, which doors were open. A banner reminds you that you are looking at the past; choose **Back to live** to return to the present. The playhead only looks back. It never sends a command; the rewound cards only show state.

Notes. Helper adds a gentle note when something looks worth a look: a light left on for 6 hours (30 minutes while every area is armed Away), a door open for 15 minutes, a temperature 4° away from the setpoint for 45 minutes, a door opened while the system was armed, an access reader that denied access, or a sensor or light that keeps changing (more than 60 changes in an hour for a zone, 30 for a light). Notes are hints, not alarms. They do not send notifications and they are not security alerts.

Scripts and the assistant. Your own Helper scripts can write the Timeline too: a note for something that happened (*Pool pump started*), or a named moment (*Movie night*) that opens when something begins and closes when it ends. A script can also quiet a chatty sensor, reword an automatic event, or take over the Timeline entirely so that only the script writes it; the lanes, offline gaps and camera events stay, and doorbell rings and alarms always show. An AI assistant connected through MCP or the HTTP API can add notes and moments the same way, never rules. There is no switch for this: a rule lasts as long as the script that set it, so disable or delete the script to stop it. Helper now also catches lights the panel changes quietly, ones it learns from a routine status check rather than an announcement; they appear with a "~" before the time, because Helper knows the change had happened by then, not the exact minute. See **MyroTimeline** in the Scripting Guide.

Doorbell pictures. When a doorbell rings and exactly one camera is in the doorbell's room, Helper keeps one picture from that camera and shows it in the *Doorbell* moment. A ring on a UniFi Protect doorbell keeps Protect's own picture of the visitor instead. Alarms, doors and panel motion never take a picture; UniFi Protect detections can, only when you allow it (below). Doorbell pictures are kept for 24 hours and then removed, whatever the window you view. The switch is in the browser's **Settings** (the gear) under **Timeline pictures**: **Keep a picture when the doorbell rings**. Turning it off deletes the doorbell pictures already kept and stops new ones; turning it on keeps pictures from the next ring. Savant homes have no panel doorbell, so they see this switch only when a UniFi Protect doorbell is linked. Control4 homes have neither, so they keep no doorbell pictures and do not see the switch. Removing a controller from Helper deletes its pictures.

Camera events from UniFi Protect. Helper can add what your UniFi Protect cameras detect to the Timeline: a person, a vehicle, an animal, a package, plain motion, or a ring on a Protect doorbell. Only cameras in Helper's camera list report (add them with **Add from UniFi Protect** above, or sync them from PRO), and nothing is recorded until you connect a console and choose what each camera reports. Myro:Home Pro and the Apple TV show the same events from Helper's Timeline. For HAI / Leviton, NESS / ELK and Savant controllers.

Connect. In **Cameras**, choose **Connect UniFi Protect for camera events** (shown when Helper has a UniFi Protect camera and no console is connected), or tick **Keep signed in for camera events** on the **Add from UniFi Protect** sign-in. The address must start with `https://`. Sign in with a local user, or with your UniFi account if it has no two-step verification; the account needs access to Protect. An account that needs two-step verification cannot stay signed in, and Helper says so instead of trying again: in UniFi OS, add a local user (**Admins & Users**, restricted to local access) and sign in with it. Helper saves the sign-in in this Mac's Keychain for that controller and console, never in a file or the log, and keeps one connection open to the console. It also remembers the console's security certificate and never sends the saved password to a console whose certificate has changed; sign in again to trust a new one. If the login keychain on the Helper Mac is locked, **Cameras** says so and camera events start once it unlocks.

Choose what each camera reports. Right after you connect, Helper lists the cameras on that console, and you choose for each one which detections become Timeline events: **Person**, **Vehicle**, **Animal**, **Package**, **Motion**, and **Ring** on a Protect doorbell. Cameras you add from the picker while keeping the sign-in, or from a console this controller already keeps, start with Person (and Ring on a doorbell); cameras that were already in Helper start with nothing until you choose. A detection the camera cannot make is greyed out, and one that is off in UniFi Protect must be turned on there too. Change the choices any time with **Camera events** on the **UniFi Protect** card in **Cameras**, in the camera's editor in the browser, or in the camera's editor in Helper's Rooms & Cameras window on the Mac. A hidden camera (**Show camera in Helper** cleared) adds no events; its choices are kept for when you show it again.

Calm by design. Person and Ring are the calm choices; vehicles, animals and plain motion can be busy. Helper folds detections close together into one event, at most six new events an hour per camera, and only a person starts a moment of its own (*Person at Front Door*, "Seen 3 times."). Vehicles, animals, packages and motion stay in the event list and the lanes. Someone who walks up and then rings becomes one *Doorbell* moment. Camera detections never change *Arrived home*, *Left home*, *Good night* or *Good morning*, and a camera with more than 30 detections in an hour gets one note that day.

One ring, not two. When a Protect doorbell rings and a panel doorbell in the same room reports the same press, Helper records one ring. If the doorbell camera is not in the panel doorbell's room (or in no room), pick the panel doorbell under **Same doorbell as** for that camera (shown once **Ring** is ticked); the default, **Automatic (the doorbell in its room)**, matches by room, and **None** keeps them apart.

Camera pictures. Pictures of a detection are off by default, because cameras may be indoors. They need both **Keep pictures when a camera sees something** under **Settings > Timeline pictures** and **Keep pictures** on that camera. Helper keeps Protect's own picture of the event (or a fresh snapshot when Protect has none) for 8 hours, with a separate allowance so camera pictures never crowd out doorbell pictures. Turning the Settings switch off deletes the camera pictures already kept. Turning off **Keep pictures** on one camera, hiding it or removing it deletes that camera's pictures and stops any picture Helper was still fetching from it. Ring pictures follow the doorbell switch above.

When Helper stops listening. Helper reconnects on its own after a network drop, sleep or a console restart, using the session it already has. It never keeps retrying a sign-in: when the console refuses the saved password, pauses sign-ins after too many attempts or asks for two-step verification, Helper stops listening at once. When the console fails to complete a sign-in, Helper tries once more an hour later and stops if that fails too. The **UniFi Protect** card in **Cameras** shows what happened, and a note appears on the Timeline in the browser, in Myro:Home Pro and on the Apple TV while camera events are paused. Choose **Sign in again** on the card to start them again.

Disconnect. Disconnect on the card stops camera events from that console and deletes the saved sign-in from the Keychain. Your cameras and your choices stay, so signing in again picks up where you left off. If the login keychain is locked at that moment, camera events stop at once and Helper deletes the sign-in when the keychain unlocks. Removing the controller from Helper also deletes its saved sign-ins and pictures.

What the timeline does not record. It shows what happened, not who did it. A light turned on from a phone, a keypad, a switch or a schedule looks the same, and Helper does not log which person or app sent a command. When the panel itself reports a user for arming or disarming, that name is added to the event. The timeline is also not an audit log or a security record; the panel's own event log, under **Events**, remains the record of what the panel saw.

Savant-only homes. A Helper that runs only a Savant system lists that controller for browser sign-in, using its Control Login or an API key, which is what lets an Apple TV pair with it. The sidebar then shows **Overview**, **Timeline**, **Rooms**, **Activities** and **Cameras**. **Activities** lists each Savant activity by room with its reported state and **Run** and **Stop** buttons; activities do not take a favorite star. There are no Security, Climate, Audio, Messages, Events or Graphs views for Savant, and Helper's native **Rooms** pane keeps the full media controls.

Control4 homes (preview). A Control4 Link controller is listed for browser sign-in the same way, using its Control Login or an API key, so an Apple TV can pair with it too. The sidebar shows the same **Overview**, **Timeline**, **Rooms**, **Activities** and **Cameras**. **Activities** lists each Watch and Listen activity by room (for example *Living Room · Watch*) with its reported state, **On**, **Off**, or **Unknown** while the room is not reporting, and **Run** and **Stop** buttons. **Stop** turns off that activity only while it is the one playing in the room. Activities do not take a favorite star. The Timeline shows each activity starting and stopping (*Living Room: Watch Apple TV on*) in the audio and media lane; volume and mute changes are not listed as events. There are no Security, Climate, Audio, Messages, Events or Graphs views, no doorbell pictures and no UniFi Protect camera events for Control4. **Whole Room Off** and Play, Pause, Stop, Next and Previous stay in Helper's native **Rooms** pane.

Normal room views show only assigned controls that Helper can operate. HomeKit assignments and unavailable Savant activities stay hidden there; they remain visible with explanations in edit mode. Control HomeKit items in PRO or Apple Home. Under **Edit room > Show controls in Helper**, clear a control to hide it only on this webpage, without removing its shared assignment in PRO or Classic. Visibility choices survive refreshes, restarts and native sync. Use **Assign controls** to change shared room membership. Savant cards work when Helper has exactly one configured system matching the saved installation and activity route, with current feedback and discovered power commands. Otherwise they show the reason they are unavailable. **Turn on** and **Turn off** affect only the selected activity. Status comes from Savant feedback; commands are not replayed after a reconnect. An authenticated browser can operate only Savant activities already assigned to its controller’s shared rooms. Paired-client sync introduces source routes; browser editors can reuse existing shared sources but cannot invent or retarget them. A new/changed browser source route returns 403 and must first be synced from paired PRO. Source identities, labels and routes require the updated PRO sync code; use PRO's Edit > Sync with Helper after updating to review existing room changes. Classic preserves those assignments when editing a shared room, even when its local schema has no matching controls.

PRO and Classic use **Edit > Sync with Helper** in Rooms or Cameras to review shared changes. Finding or reconnecting to Helper does not import, upload or remove anything, including controller photos; there is no periodic room/camera reconciliation. Tapping **Edit** makes one read-only check of `GET /helper/syncStatus`; if Helper's camera, room or photo revision differs from what this device last synced or ignored, the **Sync with Helper** button shows a badge. There is no banner, no check on entering the tab and nothing periodic. The review sheet has a direction control: **Both ways** (three-way merge; each conflicting room, camera or photo needs a choice of **Use Helper** or **Use this device**, default **Skip**, which keeps both versions), **From Helper** (only this device changes; Helper wins conflicts) and **To Helper** (only Helper changes; this device wins

conflicts). The review lists additions, updates, removals, assigned controls and photo changes before **Apply**. Conflicts never create automatic copies. If the local or shared data changes after review, review again; a race during Apply stops instead of retrying with new revisions. Some already-reviewed stages can have completed before a later stage fails, so review the remaining changes again.

Ignore means "until Helper changes again": the device remembers the ignored generations and the badge returns only after a newer one. **Turn off** and **Turn on** are in the sheet. Opening review is read-only. Controls appear only with an authenticated Helper for the selected controller, with no subscription requirement. Existing local data remains usable while Helper is unavailable. Live controller feedback is independent of layout sync.

Identity is derived, never matched by name. Panel rooms are `room-<panel number>` (number below 1000) on every device and in Helper. App-created rooms are `room-<roomGUID lowercase>` with numbers from 1000 up. Panel cameras are `setup-<number>`; app cameras use their lowercase `cameraGUID`. Helper rejects any other room ID form with 400. Panel rooms are never deleted: a delete mutation for `room-<int>` sets `webHidden: true` and keeps the layout. App rooms get tombstones, so another device cannot silently resurrect them. Helper refuses a camera create whose normalized connection (scheme, host, port, path, username; password excluded) already exists, returning 409 with that mutation ID in `conflicts`; the client adopts the existing record. In PRO and Classic, favorites, local visibility and display order stay on each device (browser favorites are kept by Helper, below). Classic does not create new empty cards for rooms containing only HomeKit/Savant controls; those assignments remain in Helper. Layout revision (`revision`) and photo revision are independent: assigning or replacing a photo changes only `photoRevision`. Myro:Home for Android 1.4.1 or later takes part in this shared room/camera sync too. Helper rejects the older automatic room/camera sync protocol with HTTP 426, and `POST /helper/rooms` (whole-list replacement) returns 410 Gone; update PRO and Classic on every participating iOS device before sharing. The catalog file is version 2; a version 1 file is migrated on first read, keeping panel-numbered and UUID rooms with photos and dropping hash-identified rooms.

Matching room and camera records establish the device's sync baseline even when the review says **Up to date** and no Apply is needed. A new room carries its photo on the first approved sync, with one photo upload per room. Photo transfers use the reviewed local image and Helper photo revision; changing either before the transfer completes requires another review rather than overwriting a newer local choice or uploading an unreviewed image.

Apply in PRO and Classic reports success or an error directly and returns the review to its status at the top. Classic previews only controls it supports; HomeKit/Savant-only layouts stay in Helper and cannot replace existing Classic room controls. Older empty unsupported imports can be hidden once through the reviewed preview.

Security control is off by default for a Control Login. Enabling it still requires confirmation and your panel code for each security action. API-key sign-in follows the key's security permission. Once you sign in, reloading the page or opening another tab in the same browser keeps you signed in, until you choose **Sign out**, quit the browser, or leave it idle for 15 minutes (2 hours at most). For a wall tablet or a computer that stays on, tick **Keep me signed in on this device** when you sign in (or turn it on in the web control's **Settings**): that device then stays signed in for 30 days after it was last used, even across browser and Helper restarts. Security actions still ask for your panel code every time. To end every sign-in for a controller, for example after losing a tablet, choose **Sign Out All Browsers** under **API > Control Login**. On a computer other people use, don't keep it signed in, and sign out when you are done. Graphs and Ethernet Quality open without signing in; the web control asks the first time you choose any other view. Favorites contain only object identifiers. They are kept by Helper for each controller, so every browser and device signed in to that controller shows the same

stars; a list that older Helpers kept in one browser is copied to Helper the first time. Choose **System**, **Light**, or **Dark** in the header; this preference is saved across Control, Graphs, and Ethernet Quality on the same Helper address.

Controller changes update the browser through a live notification channel, with periodic refresh as a fallback. **Graphs** and **Ethernet Quality** open within the signed-in Control interface so switching views does not require another login. Their standalone pages remain available without signing in, as in previous versions. Graphs load a bounded date window from an incremental index of the history files. The first request may show indexing progress. Zooming retrieves finer detail.

Thermostat history includes hourly, daily or weekly reported heating/cooling intervals and unknown coverage. Unknown is not off: long quiet periods, disabled logging and communication gaps cannot establish equipment runtime. Panel events are the panel's own event history, not Helper's diagnostic log. Use **Refresh from panel** to download recent records, or explicitly select **All available events** for a fuller read. Full reads are capped at 2,000 records and two minutes; limits or connection failures are reported rather than called a complete history. Panel event dates do not include a year.

> Browser control is for a trusted LAN or VPN. The HTTP server does not encrypt passwords, keys or panel codes. Never expose port 9393 directly to the internet. Use a unique Control Login password, not one shared with another account.

Wall and Countertop Tablets

Browser control has a tablet layout modeled on Myro:Home Pro on iPad: a rail of views down the right edge with a status dot and the Settings gear, and two panes, the view's summary on the left and its tiles on the right. Overview, Security, Rooms and Climate split across the two panes; Timeline, Control, Audio, Messages, Events, Graphs and Ethernet Quality use one wide pane. Views that do not fit on the rail are under **More**. Phones, portrait screens, Split View and desktop browsers keep the standard layout.

How the Tablet Layout Turns On

- **Automatically.** A browser uses the tablet layout when the screen is landscape, touch is its main pointer, and the window is at least 1000 × 600. Rotate to portrait and the standard layout comes back; a rotation never switches the layout while a confirmation or panel-code window is open.
- **From Settings.** In web control's **Settings**, choose **Layout**: **Auto** (the rule above), **Tablet** (always, whatever the screen) or **Standard** (never, even on a tablet). The choice is kept by that browser only. A smaller tablet that reports less than 1000 × 600 can still use the tablet layout by choosing **Tablet** here.
- **From the address.** Add `?layout=tablet` or `?layout=standard` to the web control address to force one layout, for example `http://HELPER_HOST:9393/helper/control?layout=tablet`. While the address fixes the layout, Settings shows the choice but cannot change it.

In the tablet layout, **Security** shows the zones on the left, the areas and their arm buttons on the right, and a red or green band across the top. The panel-code window has an on-screen keypad, so no keyboard covers it; a hardware keyboard still works. Every security action still asks for confirmation and your panel code.

The Wall Tablet Address

For a tablet that stays on a wall or a counter, open `http://HELPER_HOST:9393/helper/tablet` (add `?controller=` and the controller ID to skip the controller choice). It is the same web control in the tablet layout, plus wall-tablet behaviour that never reaches a personal iPad or a computer on the usual address:

- **Ambient mode.** After 5 minutes without a touch the screen dims to 75% black with the time and date, which move every 10 seconds. Live camera pictures stop while it is dim. The first tap only wakes the screen; it never presses the control under your finger. An alarm, or a zone that trips, wakes the screen and opens **Security**; no keypad opens by itself. The screen does not dim while an alarm is sounding or while a room or camera editor is open; a panel-code window left open is cancelled when it dims.
- **Overnight reload.** Once between 3:30 and 4:30 a.m., while nobody is using it, the page reloads itself for a fresh start. If status stops arriving for 2 minutes, it reloads as well (waiting longer each time if the problem continues). It reloads only when Helper answers, so a Mac that is asleep never leaves the tablet on a browser error page, and it signs straight back in.
- **Always signed in. Keep me signed in on this device** is turned on and cannot be turned off, so the tablet stays signed in for 30 days after it was last used, across reloads and Helper restarts.
- **No sign-out on the screen.** The **Sign out** button and the Myro:Home link are hidden, and so are the settings that would change things for every browser: the home photo's **Change** and **Remove**, and the Timeline picture switches. The tablet's own choices stay: the layout, the camera switch, **Show the home photo** and **Low-power display**.
- **Touch only.** No pinch zoom, no text selection callouts and no pull-to-refresh. Every 3 minutes the whole page shifts by 2 pixels so nothing stays in exactly one place on the screen.

- **Low-power display.** Under **Settings > Wall tablet**, **Low-power display** turns off the glass blur, for older iPads and Fire tablets that stutter.

Adding `?kiosk=1` to the usual address turns on the same wall-tablet behaviour without forcing the layout.

Keep the Screen Awake

A web page served over Helper's local `http://` address cannot keep a tablet's screen on by itself; the browser allows that only on secure pages. Set it on the tablet instead:

- **iPad: Settings > Display & Brightness > Auto-Lock > Never**, and keep the iPad on power. To keep it in the browser, turn on **Guided Access** (**Settings > Accessibility > Guided Access**), open the web control, and triple-click the top or Home button to start it.
- **Android and Fire tablets:** set the screen timeout to its longest, or use Fully Kiosk Browser, which keeps the screen on and the page full screen.

Add to Home Screen

Open the wall tablet address, then use the browser's **Add to Home Screen** (Safari's Share menu on iPad; **Add to Home screen** or **Install app** in Chrome). The page opens full screen from its Myro:Home icon, at the address you added it from, so add it from `/helper/tablet` for the wall-tablet behaviour. A Home Screen icon keeps its own sign-in, so sign in once after you first open it from there.

Cameras on a Tablet

Cameras are off on a tablet by default, because a shared screen shows them to anyone walking past. In **Settings**, turn on **Show cameras on this tablet** to bring back the **Cameras** view on the rail and the camera pictures in **Rooms**. The choice is kept by that browser only. While it is off, the tablet shows no camera pictures or video in Rooms and Cameras; Timeline pictures (a doorbell ring, a UniFi Protect detection) still appear when the Timeline has them.

Signing In a Tablet

A wall tablet signs in like any browser, with the controller's Control Login or an API key, and it can do what that sign-in allows: control the home and, when the sign-in has security control, arm and disarm with the panel code each time. Five wrong codes lock security control for that controller for 15 minutes. Hiding sign-out and the Helper-wide settings on the wall tablet address is a convenience of the page, not a limit on the sign-in, so treat a wall tablet like any other kept browser and sign it in only on a network you trust.

- **To sign a tablet out**, open `http://HELPER_HOST:9393/helper/control?layout=standard` on the tablet and choose **Sign out**, or choose **Sign Out All Browsers** under **API > Control Login** in Helper (this also signs out every other browser and paired Apple TV for that controller). Saving a new Control Login password signs out every Control Login session too.
- **To be able to sign out one tablet on its own**, sign it in with an API key generated for that tablet under **API > API Keys**, without **Arm and disarm** if it should not arm. Removing that key later signs out that tablet only.
- Helper keeps up to 32 kept browser sign-ins across its controllers; past that, the one used least recently is signed out and shows the sign-in page again.

Supported Tablets

- **iPad** with iPadOS 15.4 or later, in Safari or from the Home Screen. The tablet layout fits screens from 1024 × 768 up to 1366 × 1024 (the 12.9-inch iPad Pro).
- **Android tablets** with Chrome.
- **Amazon Fire tablets** with Silk or Fully Kiosk Browser.

Symptom	First Check
The tablet shows the standard layout	Hold it in landscape, or choose **Tablet** under **Settings > Layout**. A screen smaller than 1000 × 600 needs **Tablet** chosen
The screen turns off	Set Auto-Lock to **Never** (or the longest screen timeout) and keep the tablet on power; the page cannot keep it awake
The tablet asks to sign in again	It was signed out from Helper, its sign-in was not used for 30 days, or more than 32 browsers are kept. Sign in again on the tablet
Cameras are missing	Turn on **Show cameras on this tablet** in Settings
The page shows **Could not load controllers**	Wake the Helper Mac; the tablet keeps retrying and loads on its own once Helper answers

For anything else, visit www.myrocontrol.com/support.

Apple TV

Myro:Home Pro for Apple TV 2.0 gets your rooms, their controls and your recent activity from Helper, so it needs a Helper that is running on your network. Cameras come from Myro:Home Pro on your iPhone or iPad over iCloud (turn on **Show on Apple TV** for a camera there); Helper tells the Apple TV which room each camera belongs to. Favorites are kept on each Apple TV, and the first time it pairs it copies Helper's favorites to start with. A Today row lists recent activity at home from the last day, such as doorbell presses, lights going on and off, and security, door and motion events. What appears depends on your controller and what Helper has logged. Savant-only homes and Control4 homes can pair too. Support for NESS / ELK, Savant and Control4 remains a preview.

Pairing is a one-time approval on the Helper Mac. The Apple TV never needs your Control Login password.

Pair an Apple TV

1. Make sure Helper is running on the Mac, the Mac is awake, and the Apple TV is on the same home network.
2. On the Apple TV, open Myro:Home Pro, then **Settings > Myro Helper**. It finds Helper on the home network by Bonjour. Choose your controller if more than one is listed.
3. The Apple TV shows a six-digit code.
4. On the Mac, Helper shows a window titled **Apple TV “Name” wants access to Controller**, with the same code and the buttons **Allow** and **Deny**. The window stays on screen even if Helper is in the background.
5. Confirm that the code on the Mac matches the code on the TV, then click **Allow**. Click **Deny** if you did not start this request.
6. The Apple TV signs in and shows the controller's rooms.

A request that is not answered within 5 minutes is denied automatically, and you can start again from the Apple TV. Up to three requests can wait at the same time.

> **Keep the Mac's address the same.** A pairing is tied to the address the Apple TV used to reach Helper when it paired: the Mac's name or IP address, and port `9393`. If that address changes, the Apple TV will ask to pair again. This happens after a new router, a DHCP change, a switch between the Mac's name and its IP address, or a different network interface on the Mac. Give the Helper Mac a fixed IP address by making a DHCP reservation for it in your router, and keep the Mac awake and on the same network. The Apple TV must also reach Helper by the same Bonjour name every time. Do not point it at Helper through Tailscale, a VPN or another network address; if the Apple TV moves to another network or reaches Helper by a different name or address, it has to pair again.

How Long a Pairing Lasts

A pairing lasts 30 days from the last time the Apple TV used it, and it renews itself while the TV is in use. An Apple TV that is not used for 30 days has to pair again.

See and Revoke Paired Apple TVs

Select the controller in Helper, then open **API > Control Login** and choose **Paired Devices...**. Each Apple TV paired with that controller is listed, with a **Revoke** button. Revoke an Apple TV you no longer use, or one that has left the house; it has to pair again to get back in.

Sign Out All Browsers on the same tab also signs out every paired Apple TV, along with the browsers.

Arming and Disarming from the Apple TV

A paired Apple TV can arm and disarm, but only when **Allow security control (panel code required)** is turned on for that controller's Control Login in Helper (**API > Control Login**). It always asks for the panel code, every time. Five wrong codes lock security control out for 15 minutes. An Apple TV cannot bypass or restore zones and cannot force arm: if the panel will not arm because a zone is not ready, close the zone, or use the panel or the Myro:Home app. Turning the setting off applies to paired Apple TVs on their next request.

If Something Does Not Work

Symptom	First Check
The code window does not appear on the Mac	Confirm Helper is running and the Mac is awake, then start the pairing again from the Apple TV. Check that the Apple TV and the Mac are on the same network
The Apple TV does not list Helper	Put the Apple TV and the Mac on the same local network. A guest network, another VLAN, or a router that blocks Bonjour hides Helper from the Apple TV
The request is denied or expires	Answer the window on the Mac within 5 minutes. Start a new request from the Apple TV
The Apple TV keeps asking to pair	The Mac's address has changed. Set a DHCP reservation for the Mac in the router, keep using one address for it, and pair again
An Apple TV was revoked and cannot get in	Revoked Apple TVs must pair again

> Do not expose port `9393` to the internet. The Apple TV stays at home, on your local network. For your phone away from home, use Tailscale as described in *Reaching Helper Away From Home* and *Remote Access with Tailscale*.

Enable Push Notifications

Register each phone or tablet that should receive alerts from a controller:

1. In Helper, select the controller and open **Notifications > App**. Enable **Myro:Home (iOS & Android)** or **Myro:Home Pro**, as appropriate.
2. Open **Notifications > Configuration** and leave the setup QR code visible.
3. On the iOS or Android device, open the Myro:Home app and open that controller's settings.
4. Scroll to **Push Notifications**, tap the scanner icon, and scan the QR code displayed by Myro:Home Helper.
5. Repeat on every device that should receive notifications, then select **Send Test Push Notification** in Helper to confirm delivery.

Scanning registers that specific mobile device. Use **Notifications > Event Settings** and **Notifications > Object Settings** in Helper to choose which controller events and objects generate alerts.

Everyday Operation

Once configured, Helper is designed to run unattended. The **Status** tab is the quickest health check: it shows the controller connection, uptime, panel details, power state, and active troubles.

The main toolbar provides the routine tools:

- **Sync Now** refreshes names and object definitions from the controller.
- **Show Log** displays operational events and errors. Leave Verbose off for normal use; enable it when diagnosing a problem or when support requests additional detail.
- **Scripting** opens the automation editor.
- For an advanced local integration, the terminal icon in **Scripting** opens **User-Installed Scripts**. This optional 4.51 feature runs a trusted executable you install, such as a Python program, without an HTTP listener. It is off by default and runs outside Helper's sandbox once enabled. Review the permission warning and the Scripting Guide before using it; Python and its packages are your responsibility to install.
- Start external integrations with the Scripting Guide's harmless `helper-probe.sh` setup check before connecting a real device. A 60-second timeout reports failure but does not stop the external program; the guide explains how to diagnose a stuck task. This feature is optional and is not needed for Helper's built-in integrations.
- **Graphs** opens locally stored sensor history; **Ethernet Quality** is a separate browser view.
- **Control** opens browser sign-in for the selected controller.

Reaching Helper Away From Home

This section describes how the apps reach Helper itself, for HAI / Leviton and NESS / ELK controllers and for Savant and Control4 controllers paired with Myro:Home Pro. For PRO's direct Savant connection, follow **Savant Outside the House** below; learning the Helper Mac's address does not create a route to the Savant host.

Myro:Home apps find Helper by Bonjour, which only travels across a local network. A VPN such as Tailscale carries no Bonjour traffic, so an app away from home cannot discover Helper on its own — even though Helper answers on every address the Mac has, including its VPN address.

To make remote access work, **open the app once on your home network and let it connect to Helper.** During that first successful connection Helper tells the app every address it answers on, and the app remembers them. From then on, when the app cannot find Helper on the local network it tries those addresses directly, so it reconnects over the VPN without any setup.

This only works in that order. An app installed while away from home has never connected, has nothing remembered, and will report that no Helper was found. Bring it home, let it connect once, and remote access works from then on.

The addresses are only shared after the app proves it holds the controller's discovery/pairing key: the HAI private key, or the pairing code Helper issues for an ELK, Savant or Control4 controller. They are not handed to an unpaired app.

Remote Access with Tailscale

A VPN can provide private access from your phone to the Helper Mac. One option is Tailscale, which connects devices in your private Tailscale network (your **tailnet**) without forwarding Helper's port on your home router. The steps here reach Helper; direct Savant access has an additional routing step below.

To set it up:

1. Install Tailscale on the Mac running Helper and sign in.
2. Install Tailscale on each iPhone or iPad running Myro:Home or Myro:Home Pro and sign in with the same account.
3. Leave it connected on the phone. On the Mac, keep it signed in alongside the requirements in *Make the Connection Persistent* — the Mac must still be awake, logged in, and online.
4. Leave Tailscale's **Allow LAN access** setting on, on both the phone and the Mac (on macOS it is called *Allow local network access*). It only matters if you also use a Tailscale **exit node** — but with an exit node on and LAN access off, Tailscale sends everything through the tailnet: the app can no longer discover Helper when you are at home, because Bonjour only travels on the local network, and the Mac itself can lose its path to the controller at its `192.168.x` address, which disconnects Helper from the controller.

Helper needs no configuration for this. It already listens on every address the Mac has, so once Tailscale is running the Mac gains a second address — one beginning with `100.` — that reaches Helper from anywhere, and Helper reports it to your apps automatically.

Then follow *Reaching Helper Away From Home*: open the app once on your home network so it learns that address. Bonjour does not travel over Tailscale, so discovery alone will never find Helper remotely; the app relies on the address it remembered.

The same connection makes the rest of Helper reachable while you are away — the graphs page, the local HTTP API, and the MCP server all work over Tailscale using the Mac's `100.` address in place of its home address.

> Tailscale is an alternative to exposing Helper to the Internet, not an addition to it. Continue to leave port `9393` closed on your router.

Reaching the controller itself

On an HAI / Leviton or NESS / ELK system the app still talks to the **controller** directly, on its own port (`4369` for Omni-Link, `2601` for ELK). Tailscale on the phone and the Mac connects those two devices to each other; it does not by itself let the phone reach the controller's `192.168.x` address. Without that, Helper works away from home but the app cannot connect to the panel, and the usual answer, a port forward on the router, is exactly what Tailscale is meant to replace.

Make the Helper Mac a Tailscale **subnet router** so the phone can reach the home network through it:

1. On the Mac, in the Tailscale menu bar app, open Settings and turn on **Run as subnet router**, or in Terminal run `tailscale up --advertise-routes=192.168.1.0/24` using your real home subnet. Keep *Allow local network access* on.
2. In the Tailscale admin console, open that Mac's machine entry and **approve** the advertised route. It stays off until you approve it.

3. In the app, set the controller address to the controller's **LAN address** (the one you reserved on the router), not a DDNS name. The port stays as it was.
4. Test on cellular with Tailscale connected. When that works, remove the router's port forward and test again.

The Mac advertising the route must stay awake and on the home network. If you run Helper on more than one Mac, advertise the route from **only one** of them, the one that never leaves the house; a laptop that advertised the route would carry the panel's traffic into whatever network it visits. Every other Mac, phone and iPad simply joins the same tailnet, and each Helper reports its own `100.` address to the apps.

Every device in the household signs in with the same Tailscale account or is invited to it; nothing is configured per device beyond installing Tailscale.

Savant Outside the House

PRO connects directly to the Savant host for rooms, activities, and live media status, with Helper optional. Away from home, your phone needs a VPN route to that host. Installing Tailscale on the Helper Mac makes **Helper** reachable; it does not automatically relay PRO's Savant connection or make the Savant host's LAN address reachable.

For a Savant host on your home LAN, use a **Tailscale subnet router**: an always-on device that advertises a route to that LAN. It can be the Helper Mac if separately configured for subnet routing, or another supported home gateway. You do not need to install software on the Savant host. Follow Tailscale's subnet-router setup for the gateway's operating system.

1. **Set up Savant at home first.** Verify PRO connects, trust the correct host identity/certificate, and sign in with a Savant **Local User**. Record the host's LAN IP address and keep it stable, for example with a DHCP reservation. The Local User password is not the Mac mini administrator password.
2. **Advertise the correct home subnet.** For example, a host at `192.168.50.24` might be on `192.168.50.0/24`. These are examples: use your actual network address and subnet mask, not a guessed `/24`.
3. **Approve the route and permit access.** Approve the advertised subnet in Tailscale's admin console and check that its access rules allow your phone to reach the Savant host. Route approval and access permission are separate settings. Keep the gateway online and able to reach Savant locally.
4. **Connect the phone to the same tailnet.** In PRO's Savant setup, use the host's routed **LAN IP address**. A `.local` name or automatic Discover may not work remotely. If entering an address, verify the host and confirm the expected identity/certificate. Continue using the Savant Local User; Tailscale provides network reachability, not Savant sign-in.
5. **Test away from Wi-Fi.** With cellular data and Tailscale connected, open PRO and check the room list and reported activity, volume, and mute. If you choose to test a command, confirm both the physical result and subsequent live feedback. A successful connection to Helper alone does not validate the Savant route.

The Savant Controller Guide lists the connection services required by this release. A routed connection must carry discovery to the configured address, inventory/photos, and the authenticated media/state session. Home-network broadcast discovery does not extend across a normal Tailscale connection, so use a known host address. An exit node or **Allow LAN access** setting alone does not configure the subnet route described here.

Symptom away from home	Check
Helper is reachable, but Savant is offline	The subnet route, route approval, access rules, and Savant host address; Helper is not a direct-media proxy
Discover finds nothing	Enter the known routed LAN IP and verify the host
Connection works, but rooms or photos do not load	The host's local inventory/resource service must also be reachable
It works on cellular but not another Wi-Fi network	Check for overlapping home/remote address ranges or another VPN; consult Tailscale's routing guidance
Host identity or certificate differs	Verify the intended system before accepting a replacement

Keep Savant services and Helper's port `9393` off the public Internet. Local Helper scripts can continue monitoring Savant and controlling configured integrations while your phone is away; that is separate from PRO's direct room controls. This documents the required network path; validate your particular VPN and Savant installation using the cellular test above.

Connecting at Home with an Outside Address

If your apps reach the controller through an outside address (a DDNS name or a port-forwarded public IP), some home networks cannot connect to that address from the inside; this is common with double NAT and mesh routers. The apps always try the address you configured first. If that cannot be reached, they fall back to your controller's **LAN address**, which they learn from Helper 5.20.

Helper shares only a private LAN address (such as `192.168.x.x`) that it is actually connected to. It never shares a public address, a DDNS name or a Tailscale address. So configure Helper with the controller's LAN address, or a name that resolves to it locally, and it will have a LAN address to share. You can confirm what an app learned under **About > Controller LAN address**.

Helper is set up with	What the app learns	At home, when the outside address fails
The controller's LAN IP	That LAN IP	The app falls back to it and connects
A DDNS name that works from inside (hairpin NAT)	The LAN IP Helper connected to	The app falls back to it and connects
A public IP or a DDNS name that Helper cannot reach locally	Nothing	No fallback; change Helper to the LAN address
The app already uses the LAN IP	The same LAN IP	Nothing changes; it connects directly

When Something Does Not Look Right

Symptom	First Check
Controller will not connect	Verify the address, credentials and port specified by your controller guide, plus local network reachability
Names or objects are missing	Confirm the controller is connected, run **Sync Now**, and wait for the progress overlay to finish

Symptom	First Check
A mobile app cannot find Helper	Put both devices on the same local network and check private Wi-Fi address and firewall rules. Away from home, see *Reaching Helper Away From Home* — the app must have connected on the local network at least once
Helper or the app stopped finding things at home after turning on a Tailscale exit node	Turn on Tailscale's **Allow LAN access** (*Allow local network access* on macOS) on that device, or turn the exit node off. See *Remote Access with Tailscale*
HomeKit does not show a change	Confirm the object is HAP-enabled and restart the HAP server after configuration changes
A script or integration is not responding	Open **Show Log**, reproduce the action, and look for the controller or integration name near the error
Graphs are empty	Enable logging/graphing for the desired sensor and allow new readings to arrive
Helper keeps coming back after a Force Quit	That is the restart-after-crash behaviour; a Force Quit cannot be told apart from a crash. Quit Helper normally, or turn off **Start Myro:Home Helper automatically when logging into this Mac**

> Do not expose the controller or Helper's port `9393` directly to the public Internet. Use them on a trusted local network or through a properly secured remote-access solution.

Learn More

The Help menu offers shared guides and a separate guide for each controller family:

- **Getting Started** for initial setup and daily operation
- **MCP Guide** for connecting an AI client, choosing API-key permissions, and using MCP tools safely
- **Scripting Guide** for shared Helper APIs, integrations, script management and schedule syntax
- HAI and Leviton Controller Guide for HAI setup, panel APIs/events, solar behavior and HaikuHelper migration
- NESS / ELK Controller Guide for ELK setup, panel APIs/events, solar location and limitations
- Savant Controller Guide for Local User sign-in, rooms, live feedback, media controls and cross-controller scripts
- Control4 Controller Guide for Control4 Link setup, room activities, Apple Home switches and scripting
- **HTTP API Documentation** for local API authentication, shared reads, scripts, graphs, and discovery

For product support, visit www.myrocontrol.com/support.