Myro:Home Helper

Control4 Controller Guide

Version 5.50

Control4 Link setup, room activities, Apple Home switches, and scripting through an installer-loaded connector.

Myro Control, LLC | myrocontrol.com

Contents

Connect your Control4 system

- Preview status and requirements
- Set up the connection
- Activities and state

JavaScript API Reference

- Rooms pane, HTTP and MCP
- Apple Home
- Frequently asked questions
- Diagnostics and updates

Connect your Control4 system

Control4 Link 1.1.0 | Protocol 1.1. Control4 Link connects Helper to an installer-loaded Myro DriverWorks connector. The local connection uses a unique installation credential and pinned TLS certificate. Helper publishes selected room activities to Apple Home and exposes scoped scripting. Your existing Control4 project remains responsible for equipment control, routing and automation.

Preview status and requirements

Control4 Link is an **installer and field-testing preview** in Helper 5.50 and later. Actual Control4 hardware acceptance is still pending. Downloading the driver does not add Control4 support to an older Helper app.

- **Mac:** macOS 12 or later, Apple silicon or Intel. Keep the Mac awake with Helper running on the home LAN.
- **Connector:** Myro Control4 Link 1.1.0, protocol 1.1 (requires Helper 5.50 or later, which understands the connector's explicit "no source selected" value). The driver declares Control4 OS 4.3.0 as its minimum. CORE 5 on OS 4.3.0 with Halo is the initial compatibility target; this is not a claim of completed hardware testing or support for other controllers.
- **Commissioning:** owner-authorized access to compatible Windows Composer Pro, a project backup and the actual room/source IDs. Python 3.9+ and OpenSSL 1.1.1+ or 3.x run on the installer workstation, not the Control4 controller. Windows or RDP is needed for commissioning and maintenance, not ongoing operation.
- **Local connection:** pinned TLS and a per-installation credential, normally on TCP 52645. The connector accepts one authenticated Helper client at a time. Do not forward this port to the public internet.

See the preview requirements and installer downloads and the preview discussion. Installer packages and this bundled Helper guide have separate version numbers.

Set up the connection

1. Have the installer back up the project, install the versioned Myro Control4 Link driver, and apply the reviewed room/source configuration.
2. Click **+** beneath Helper's controller list and choose **Control4 Link**. In **Connection**, choose **Import Trusted Connection...** and select the private `helper-connection.json` supplied for that installation. Check the host, installation identity and certificate fingerprint with the installer.
3. Choose **Connect** and verify the room inventory and observed status. Keep credentials in this Mac's Keychain when using saved connections. The JSON contains a secret; do not post it or include it in support logs.
4. Open the controller's **HAP** tab, select activities and optional **Custom Names**, choose **Apply**, then pair in Apple Home with the displayed code. Activities appear as switches. The bridge retains its identity during local connection outages.
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**. In Pro, open the controller's settings, choose **Pair with Helper** and scan the QR code or type the code. **Regenerate...** issues a new code, after which every phone must pair again. The code is for Helper features only; it is separate from the installer's connection file and never connects Pro to the Control4 system.

Control4 app account credentials, Connect subscriptions and Composer activation are separate from this local connector. The driver does not provide Halo microphone/Siri audio or change the Apple TV driver's requirements. Helper must remain running on the home LAN for its HAP bridge and persistent scripts.

Activities and state

An activity identifies a room plus a Watch/Listen mode and source proxy ID. Names are labels; use IDs from `rooms()` and `activities(room)` for saved automations. A source in an equipment room can be available in another room. The installer-reviewed connector inventory defines what Helper can control.

Feedback describes Director's selected route and reported room state. It is not independent confirmation that a projector produced a picture or an IR-controlled receiver physically turned on. Each room has a Watch path and a Listen path. A path reports the selected source ID, `"0"` when Control4 reports no selected source on that path, or nothing when the value is unknown. A Watch switch in a room that is playing audio only therefore reads Off; unknown fields remain unavailable, and a disconnected/stale activity shows No Response in Apple Home rather than a fabricated OFF.

Room Off is broad. `MyroControl4.off(room)` intentionally shuts down the whole room. Activity-specific OFF turns the room off only when the installer has enabled the room's combined shutdown policy, that activity is the selected source on its path, and the other path is not carrying a different source (the same source, no selection, or no feedback variable for that path are all fine). A stale OFF for an old source does not shut down a newer Halo selection. A different source active on the other path makes activity OFF unavailable; an unknown other path asks for fresh feedback first. Use the explicitly named Room Off action when the whole room should shut down. A rejected Apple Home write is recorded in Helper's log with the room, activity, action and error code, and the switch returns to its real state.

JavaScript API Reference

The `MyroControl4` global belongs to the script's selected Control4 controller. From another controller use `MyroControllers.get(exactGUID).control4`, with Send To permission at the origin and Accept From at the destination. Every operation rechecks grants; retaining a handle does not retain permission. Credentials and the private connection token are not exposed to scripts.

Method	Behavior
`rooms()`	Cached room inventory, IDs, names and activities.
`activities(room)`	Cached allowed Watch/Listen activities for the exact room ID or unique name.
`status(room)`	Observed room state/freshness; unknown or unavailable state can return null/omit fields. `watchSourceID` / `listenSourceID` are a source ID, `"0"` for no selection, or omitted when unknown.
`start(room, activity)`	Select the exact allowed room activity through Control4 routing.
`off(room)`	Explicit whole-room shutdown.
`volume(room, level)`	Integer level 0 through 100, when supported by room feedback/control.
`mute(room, muted)`	Boolean explicit mute/unmute when supported.
`play(room)`, `pause(room)`, `stop(room)`	Supported media commands through the selected room route. Stop is playback stop, not power off.
`next(room)`, `previous(room)`	Supported next/previous media commands.
`transport(room, activity, action)`	Explicit activity targeting for `play`, `pause`, `stop`, `next`, `previous`, or guarded `off`.
`ScriptEvents.onControl4RoomChange(roomOrStar, handler)`	Observe room changes; handler receives roomName and state. Initial/reconnect snapshots establish a baseline without firing automations. Unknown/stale/offline feedback remains unavailable, not OFF.

IDs win over names. A name must uniquely match one room/activity; ambiguous names reject. `true` from a command means validated and queued, not physical success. Read observed feedback separately. The manager reports command errors/receipts through Helper's logs, without exposing secrets. Do not retry writes automatically on a lost receipt or reconnect.

- `rooms()` returns `[{id, name, activities}]`; each activity is `{id, name, mode, sourceID, commands}`, where `mode` is `"watch"` or `"listen"`, `id` is `mode:sourceID` (for example `watch:115`), and `commands` lists what the connector allows for it (`start`, `off`, `play`, `pause`, `stop`, `next`, `previous`). `activities(room)` returns that room's activity array, or `[]` for an unknown or ambiguous room.
- `status(room)` returns `{revision, fresh, roomID, roomName, power, watchSourceID, listenSourceID, volume, muted, updatedAt, inventory}`, or null when the room is unknown or

ambiguous or no state has been reported for it yet. `watchSourceID` and `listenSourceID` are a device ID when a source is selected, `"0"` when Control4 reports no selection on that path, and omitted when unknown; other unknown values are omitted. `inventory` is `{available, refreshing, activities, updatedAt}`.

- A command returns `false` at once when the script is not running on a Control4 controller (or through a permitted `.control4` handle), Helper is not connected and signed in, the room or activity is missing or ambiguous, `volume` is not a whole number from 0 through 100, or `transport` gets an action other than `play`, `pause`, `stop`, `next`, `previous` or `off`. The connection, room and activity cases also write a log line.
- After `true`, Helper or the connector can still refuse the request, and the refusal is logged: room state is not fresh yet, the activity does not list that command, 64 commands are already awaiting feedback, or the room changed before the command arrived. Room-level `play` / `pause` / `stop` / `next` / `previous` need the room on with exactly one active path: a path with no selection (`"0"`) does not count, two different sources do; use `transport(room, activity, action)` otherwise. `transport` media actions need that activity to be the active one. `transport(room, activity, "off")` on an activity that is already off sends nothing.
- `ScriptEvents.onControl4RoomChange` returns false on a non-Control4 controller, for an unknown or ambiguous room, for a handler that is not a function, or after 64 Control4 room handlers on this controller. A change fires only while connected and only with fresh state; the handler receives the room's name and the same state object as `status(room)`, without `inventory`.

```
// First inspect inventory; substitute real stable IDs before deployment.
var rooms = MyroControl4.rooms();
console.log(JSON.stringify(rooms));
console.log(JSON.stringify(MyroControl4.activities("42")));
```

```
// One deliberate action, using installer-reviewed IDs.
MyroControl4.start("42", "watch:115");
```

```
// Whole-room shutdown. Do not confuse this with stopping playback.
MyroControl4.off("42");
```

Observed room changes

`ScriptEvents.onControl4RoomChange(roomOrStar, handler)` subscribes on the script's own Control4 controller. Pass a stable room ID, unique room name, or `"*"`. The handler receives `(roomName, state)` for observed changes. Initial connection/reconnect snapshots establish a baseline and do not fire change automations. Disabled, deleted, stopped or replaced script runs lose their handlers.

```
ScriptEvents.onControl4RoomChange("42", function(roomName, state) {
    console.log(roomName + ": " + JSON.stringify(state));
});
```

```
// Cross-controller example: only with bilateral authorization.
var target = MyroControllers.get("EXACT-DESTINATION-CONTROLLER-GUID");
if (target) {
    console.log(JSON.stringify(target.control4.rooms()));
}
```

Do not build feedback loops that issue a command each time the same state is observed. Use changes relevant to the automation and verify freshness. An initial snapshot, a transport receipt, and a physical device confirmation are distinct signals.

Rooms pane, HTTP and MCP

The native **Rooms** view lists the imported rooms with their observed state. **Start Activity** selects the chosen Watch/Listen activity, **Stop Selected Activity** stops only that activity, **Whole Room Off** shuts down the room, and Play, Pause, Stop, Next and Previous follow the commands the connector reports for that activity. Control4 has no panel objects or security, so panel routes do not apply to it. Helper's browser control (sign in with the controller's Control Login or an API key) offers a Control4 controller **Overview**, **Timeline**, **Rooms**, **Activities** with **Run** and **Stop**, and **Cameras**, and an Apple TV can pair with it; the Timeline records each activity starting and stopping. **Whole Room Off** and transport controls stay in this native pane. See **Control4 homes** in the Getting Started guide and **Control4** under **Browser Control** in the HTTP API guide.

`GET /helper/{controllerID}/status` for a Control4 controller returns `controllerFamily: "Control4"` and `backend: "control4"`, with the connection state, `rooms`, `roomStates`, `inventory`, `homeKit` (bridge settings and state, without pairing credentials), `objectCounts` (`Rooms`, `Activities`) and `capabilities`: `rooms`, `av`, `scripting` and `hap` are true; `security`, `panelCommands`, `energyCost` and `ethernetQuality` are false. In a script, `MyroController.getStatus()` reports `backend: "control4"` with `capabilities` of `media` and `hap` true and `panelSecurity`, `panelSensors` and `omniAudio` false. MCP assistants work with Control4 through the scripting reference and the script tools, using the exact `MyroControl4` methods above.

Apple Home

Select only activities you want to expose. Custom names do not change accessory identity. Removed or unavailable targets remain unavailable rather than silently targeting something else. A changed connector/project identity needs reviewed reimport/reselection. Pairing reset affects this Helper bridge and should not be a routine reconnect step.

If the same Control4 activity is already exposed through another HomeKit bridge, select one exposure path to avoid duplicate accessories. The current interface uses activity switches, matching Helper's Savant presentation; it is not a native Television input picker.

Frequently asked questions

What can Siri control?

Selected Watch and Listen activities appear in Apple Home as switches. Siri can turn those switches on and off, and they can join Home scenes and automations. Changes reported by Control4 remotes update Helper and Apple Home. The activity switches do not expose playback, volume or navigation commands to Siri.

Supported media controls remain available in Helper and scripts; Apple's native Siri media features are separate.

Does Control4 have Timeline or browser control in Helper?

Yes, since Helper 5.50. Control4 Link homes get the browser view, signing in with the Control Login or an API key, with Run and Stop for Watch and Listen activities, and a Timeline of when each one started and stopped. An Apple TV can pair through the same sign-in. Myro:Home Pro does not show a Control4 Timeline. **Pair with Myro:Home Pro** on the Connection tab is a different thing: it lets Pro use Helper's room photos for the controller and find the Helper on the network. A home's separate HAI, ELK or Savant controller keeps its own capabilities; do not assume Control4 activity is included in another controller's Timeline.

What does it cost?

The driver is free and Helper is a paid Mac app with no Helper subscription. Installer labor, third-party drivers, source services and Control4's own Apple integration may have separate requirements or charges. The local connector does not authenticate through Control4 Connect or 4Sight, and does not replace those services' requirements.

Why does an activity show No Response or refuse to turn off?

Helper uses observed Control4 feedback. A disconnected or stale activity stays unavailable rather than being treated as off, while a path that Control4 reports as having no selection reads Off. Check the local connection and current inventory. Activity OFF requires the exact current source and the installer's reviewed shutdown policy; an old activity cannot shut off a newer source, and a different source active on the room's other path makes activity OFF unavailable. When Apple Home's Off is refused, the switch returns to its real state and Helper's log records the reason. **Whole Room Off** deliberately affects the entire room and is a separate action.

Is the installer download page private?

No. The preview page and packages are public by URL, with search indexing excluded. Never upload the private connection JSON, credentials, keys or pairing data there or in the forum. Share diagnostic details without those secrets through Myro Control support.

Diagnostics and updates

Save the driver version, Helper version, controller OS/build, room/activity IDs, timestamps, observed state and exact error code. Exclude the token, private key, provisioning JSON and HAP pairing data. Use the installer package's read-only TLS diagnostic with Helper disconnected, since the connector accepts one authenticated client at a time.

Update the versioned driver in place and preserve provisioning identity and Helper HAP storage. For ordinary source changes update the reviewed allowlist without replacing installation/project IDs, key or token. Certificate/credential rotation requires a new private connection file and explicit reimport. See the installer guide for generation, deployment and rollback.