Apple TV — Control4 Driver

DTI Automation · driver file `appletv_drivermate.c4z` · sold through DriverCentral

Native Apple TV control for Control4. This driver talks to the Apple TV over the same connection the Apple TV Remote in iOS Control Center uses — directly from the Control4 controller.

Contents

- What this driver is
- Requirements
  - Control4
  - Apple TV
- Pairing with the Apple TV
  - Things that go wrong here, and why
  - Re-pairing, and moving to a different Apple TV
- Properties
  - Cloud Settings
  - Connection
  - Device Status (read only)
  - Apps and App Switcher
  - Support and Troubleshooting
- Actions
- Programming commands
- Apps, deep links and the App Switcher
  - How a mini-app tile launches an app
  - App Deep Links format
  - Installed Apps and Selected App Id
  - Deep links: opening a title
  - Passthrough Mode
  - Deep Link Region
- Fast Forward, Rewind and subtitles
  - Holding Fast Forward or Rewind
  - Fast Forward Mode
  - Fast Forward Step Interval
  - Subtitles (CC)
  - The remote in the Control4 app
  - Play only resumes what is paused
- Search
- Programming with the Apple TV's own state
  - Play, Pause and Stop
- Programming examples
- Troubleshooting
  - Nothing connects
  - Pairing
  - Buttons and commands do nothing

    - App tiles and launching
    - Volume
    - Getting a log for a support ticket
- Licensing and support
    - Before you contact support

What this driver is

A native Apple TV driver for Control4. The driver runs on the Control4 controller itself and controls the Apple TV directly over the network.

There is nothing else to install. No bridge, no helper application, no Raspberry Pi, NUC or other always-on box, no cloud relay, and no vendor account beyond the DriverCentral licence itself.

What it does

- **Finds Apple TVs for you.** One action lists every Apple TV on the network with its name and address. Pick one and the driver fills in everything it needs to connect.
- **Can follow an Apple TV that moves.** The driver remembers the identity the Apple TV it is paired to announces on the network — the same announcements **Find Apple TVs** uses, with AirPlay on. If a DHCP lease changes and the Apple TV comes back on a different IP address, the driver can recognise it at its new address and reconnect, with no re-pairing. Where those announcements do not reach the controller it leaves the address alone and says so, so a DHCP reservation is still recommended.
- **Full remote control.** Every button of the Apple TV remote, plus transport control, from Navigator, from a touchscreen and from programming. Holding **Fast Forward** or **Rewind** scans through a film the way the Siri Remote does, and a **CC** button switches subtitles on and off.
- **Launches apps.** 25 app tiles a dealer can assign, so Netflix, Disney+ or anything else the customer has installed is one press away in Navigator.
- **Opens a title, not just an app.** Given a link to a specific film or episode, the driver opens that title directly in the right app.
- **Reads the installed app list off the Apple TV,** so a dealer picks apps from a list instead of hunting for identifiers.
- **Searches the Apple TV from the Control4 app. Apple TV Search** is a search box with a remote under it: what you type appears on the television as you type, and the arrows and **OK** pick the result you want - without picking up the Siri Remote.
- **Reports what it knows:** power state, connection state, the name, hardware address and tvOS version of the Apple TV it is attached to, and a plain sentence in Last Error whenever something does not work.
- **Pairing is done once, at the device.** The resulting credential is held in Director's own protected storage, not in a driver property: it is not shown on any Composer tab, and this driver has no action that displays or exports it. The only way to remove it is **Clear Pairing**.

The driver presents three proxies from a single file: the Apple TV itself as a **Media Player**, an **App Switcher** carrying 25 app slots, and **Apple TV Search** (a search box with a remote under it).

Requirements

Control4

ITEM	REQUIREMENT
Controller OS	**4.0.0 or later.** The driver uses networking features Control4 introduced in OS 4.0.
Network	The controller must be able to reach the Apple TV over **IPv4**. For the driver to find Apple TVs by itself, the controller and the Apple TV must be on the same network — discovery traffic is commonly not carried between separate VLANs or guest networks. If they are separated, the address and port can be typed in by hand instead.

Apple TV

The driver **does not check the Apple TV's model or its tvOS version, and does not refuse any of them.** What it needs is an Apple TV that offers the standard pairing flow — the one that puts a four-digit code on the television.

Set Sleep After to Never. On the Apple TV, open **Settings** → **General** → **Sleep After** and choose **Never**. An Apple TV that goes to sleep can leave the network, and while it is gone the driver cannot reach it: Room On sends it a Wake-on-LAN packet, but many Apple TVs ignore one in deep sleep.

The driver reports the Apple TV's **name**, **tvOS version** and **hardware (MAC) address** on the Properties tab, so you can see exactly which box a driver is attached to without walking to the television. The name and the tvOS version are what the Apple TV announces on the network: they are filled in when you pick it, and again after the driver connects. The hardware address is what the controller itself sees once the Apple TV has connected.

A **DHCP reservation or a static address** is still recommended for an Apple TV. Reserve the address **Apple TV MAC** shows once it matches the **Ethernet Address** or **Wi-Fi Address** the Apple TV shows under **Settings** → **General** → **About**. Before version 1091 this field showed an identifier the Apple TV announces, not the address the controller sees: remake any reservation made from an older value.

The driver can also follow the Apple TV to a new address when the controller hears its network announcements — the same ones **Find Apple TVs** uses, with AirPlay on. Otherwise it leaves the address alone and says so.

Pairing with the Apple TV

Pairing is done **once**, at the device. The driver finds the Apple TV for you.

On a DriverCentral build, none of this works until the licence is valid. Find Apple TVs, **Pair with Apple TV** and **Enter Pairing Code** are refused along with everything else, and Last Error names the one you pressed. See Licensing and support.

A driver with no address looks for one by itself, once, when it loads — so on a fresh install the list is often already filled in before you get here. It does not browse again on later loads; a driver that already has an address has nothing to gain from it.

1. **Actions → Find Apple TVs.** Wait a few seconds.
2. **Properties → Discovered Apple TVs** — pick the Apple TV from the list. Each row shows the Apple TV's name and its address, so two boxes with the same name can still be told apart. Choosing one fills in its address, its port, its name and its tvOS version, begins connecting, and — if this driver is not paired yet — asks the television to put a **four-digit code** on screen. **Apple TV MAC** fills in a few seconds after the driver connects, unless a router separates the controller from the Apple TV.
3. **Properties → Pairing Code.** Type the four digits the television is showing. The field clears itself and pairing finishes.
4. Pairing Status goes to **Paired** and the driver connects.

Picking an Apple TV you are already paired to does not re-pair it. It re-reads the address and port and reconnects, which is the right thing to do if a device has moved, and it leaves the pairing alone.

The Actions still work if you prefer them, or if the code is no longer on screen: **Pair with Apple TV** asks for a fresh code, and **Enter Pairing Code** accepts it. They do exactly what the two properties above do.

If no Apple TVs are found, that is nearly always the network rather than the Apple TV. The traffic discovery uses is commonly not carried between VLANs, between wired and guest wireless, or across some managed switches. Put the controller and the Apple TV on the same network, or fill in **Apple TV IP Address** and **Control Port** by hand and carry on from step 3.

You do not normally need to know the port. It is not the same on every Apple TV, so the driver reads it from the device rather than asking you to find it. Every Apple TV seen so far uses `49153` and the shipped default matches that, so in practice this field is usually already correct. It remains editable for the case above.

Things that go wrong here, and why

- **Press Pair once.** If the Apple TV refuses an attempt it asks for a short wait, and the driver enforces it. Pressing Pair again inside that wait restarts the device's own back-off and is the fastest way to make pairing impossible. The driver tells you how many seconds are left.
- **Type the code while it is on screen.** A pairing nobody completes is abandoned by the driver after five minutes, Pairing Status says so, and you simply run Pair with Apple TV again.
- **Four digits, nothing else.** A letter or a longer string is refused before it reaches the Apple TV.
- **Walked away mid-pairing?** Run **Clear Pairing** to end it immediately rather than waiting for the driver's own backstop.

Re-pairing, and moving to a different Apple TV

Run **Clear Pairing**, change **Apple TV IP Address**, then pair again. Clear Pairing forgets the stored credential; there is no undo and no import, so only run it when you intend to pair again.

Properties

They appear in Composer in the order below. **Cloud Settings comes first**, so the licence state is the first thing you read on a driver that does nothing without one.

Cloud Settings

Added by the licensing clients, not part of the driver's own configuration. This driver carries **two** licence systems and **either one satisfies it**: a valid DriverCentral licence or a valid DTI activation is enough, and you only need the one that matches where you bought the driver. See Licensing and support.

PROPERTY	TYPE	WHAT IT IS
Cloud Status	Text, read only	The DriverCentral licence/account state, in DriverCentral's own words.
Automatic Updates	`Off` / `On`	Whether DriverCentral may update this driver automatically.
Activation Key	Text	Where a **DTI** licence key is typed. Leave it empty on a DriverCentral install. The key is never written to the log, and only its last four characters are stored on the controller.
Activation Status	Text, read only	The DTI activation state: `Not activated`, `Checking...`, `Checking the key with DTI...` (a key was typed and is being checked, or will be the moment a check already under way finishes), `Trial running` or `Trial – N hour(s) left`, `Activated` (with the date it was last confirmed), `Trial ended – activation key required`, or `Activated for a different controller`. That last one means the project was restored onto different hardware — a DTI activation is bound to the controller, so it needs a key issued for the new one. A status ending `the licence key could not be checked yet – retrying` means the key was typed but DTI could not be reached: the driver keeps trying, and a running trial carries on meanwhile.

Activation Method — set this first. This driver is sold through two channels and carries the licence system for both. The property at the top of the tab says which one you bought, and only that channel's rows are shown:

ACTIVATION METHOD	THE TAB SHOWS
DriverCentral *(default)*	Cloud Status, Automatic Updates
DTI	Activation Key, Activation Status

A DriverCentral dealer changes nothing. A dealer who bought from DTI Automation sets it to **DTI** once, and the Activation Key field appears.

Choosing wrongly does not break anything. The setting decides what you *look* at, never which licence the driver honours — a valid licence from either system satisfies it whatever this is set to. If the rows look wrong for what you bought, change it; nothing else needs doing and nothing needs restarting.

The two trials are measured differently, and only one is ever on screen. Both are 30 days, but not by the same clock.

- **DriverCentral's** runs on the **calendar** and is reported by their own hub in **Cloud Status**. Thirty days after it starts it is over, whether or not the controller was switched on.
- **DTI's** is measured by this driver against controller **uptime** — the hours the controller has actually been running — and reported in **Activation Status**. It is deliberately not the wall clock, because a wall clock can be set backwards and uptime cannot.

So a controller that runs eight hours a day burns a 720-hour DTI trial in about ninety calendar days, while DriverCentral's thirty days pass regardless. The DTI figure does not *pause* when the controller is off — it never counted that time at all.

Connection

PROPERTY	TYPE	DEFAULT	WHAT IT IS
Discovered Apple TVs	List	*(empty)*	Every Apple device that answered the **Find Apple TVs** action, shown as its name and address. HomePods and Macs can answer too, so pick the Apple TV. Picking one fills in its address, port, name and tvOS version, connects, and asks for a pairing code if this driver is not paired yet. Empty until the action has been run.
Pairing Code	Text, masked	*(blank)*	The four digits the television is showing. Type them here and pairing finishes; the field clears itself afterwards. Needed once per Apple TV.

PROPERTY	TYPE	DEFAULT	WHAT IT IS
Apple TV IP Address	Text	*(blank)*	The IPv4 address of the Apple TV. Normally filled in by **Find Apple TVs**. Changing it disconnects and redials immediately.
Apple TV Name	Text, read only	*(blank)*	The name the Apple TV is called on the network, as its owner set it. Filled in when you pick it, and again after the driver connects.
Apple TV MAC	Text, read only	*(blank)*	The Apple TV's hardware (MAC) address as the controller sees it on the network, read each time the driver connects. Room On sends its Wake-on-LAN packet to this address. Use it for a DHCP reservation once it matches the **Ethernet Address** or **Wi-Fi Address** the Apple TV shows under **Settings** → **General** → **About**. Blank until the Apple TV has connected at its current address, and usually blank when a router separates the controller from the Apple TV. Before version 1091 this field showed an identifier the Apple TV announces instead: remake any reservation made from an older value.
tvOS Version	Text, read only	*(blank)*	The tvOS version of the paired Apple TV, as it reports it itself every time the driver connects (from version 1092) — so it fills once the Apple TV is paired, whether you picked it or typed its address. Before the first connection it shows the version the Apple TV announces on the network, when you pick it. Nothing else depends on it.

Device Status (read only)

The driver writes all seven of these. Nothing here needs to be set, and these are the fields to read when logging is off.

PROPERTY	WHAT IT TELLS YOU
Pairing Status	`Not Paired`, `Pairing`, `Paired`, or a sentence naming what stopped it. `Paired` is the state the driver runs in.

PROPERTY	WHAT IT TELLS YOU
Connection Status	`Offline`, `Connecting` or `Connected`. While a connection is being established it may also read `Verifying pairing`, `Starting session`, `Failed` or `Working`, each followed by a growing run of dots that proves the driver is still moving.
Screen State	What the Apple TV says its screen is doing: `asleep`, `screensaver`, `idle` (the home screen) or `awake` (an app is in use). The Apple TV reports this itself, so it moves when somebody uses the **Apple Remote** as well as when this driver sends something. Some Apple TVs (seen on tvOS 26.6) report the home screen as `awake`, the same as an app.
Power State	`on`, `off`, or `unknown` — all lower case. (On a driver that has never connected it reads `Unknown`, the capitalised value shipped in `driver.xml`; the runtime only ever writes the lower-case form, so match case-insensitively in programming.) Set from events the Apple TV pushes, and by the **Refresh Status** action. The Apple TV is reported `off` only when it reports itself asleep — a screensaver, an idle home screen and an app in use are all `on`. Goes back to `unknown` when the session ends, deliberately: a dropped socket is not a television that turned off.
Last Launched App	The app this driver last *asked* the Apple TV to open. **It is not what is on screen now.** Nothing tells this driver when somebody leaves an app or presses Home on the Siri Remote. Cleared when the session ends.
Last Error	The most recent failure, in the words of whatever reported it — often the Apple TV's own error code and domain. **Blank is the pass.** Cleared on every successful connect. This is the single most useful field on the tab, because it needs no logging at all to be read.
Driver Version	`1.0.0 (build NNNN)`. Must match the build you just installed.

Apps and App Switcher

PROPERTY	TYPE	DEFAULT	WHAT IT IS AND WHAT TO SET
App Deep Links	Multi-line text	*(blank)*	Optional per-slot overrides. See App Deep Links format. Usually left blank — the driver fills it in for you.

PROPERTY	TYPE	DEFAULT	WHAT IT IS AND WHAT TO SET
Installed Apps	Dynamic list	*(empty)*	Every app the Apple TV reports installed, refreshed each time the driver connects. Pick one and its bundle identifier appears in **Selected App Id**. Empty until the driver has a session — press **Connect Now** or **List Apps** if it is. Selecting a row launches nothing.
Selected App Id	Text, read only	*(blank)*	The bundle identifier of whatever is picked above, ready to copy.
Passthrough Mode	`Off` / `On`	`Off`	What happens to the room after a mini-app launch. See Passthrough Mode.
Send Power OFF/Sleep	`Yes` / `No`	`Yes`	What **Room Off** does to the Apple TV. **Yes** puts it to sleep. **No** sends the Apple TV nothing at Room Off, leaving it awake on its screensaver, ready the next time the room is used.
Fast Forward Mode	`Held Press` / `Remote Presses` / `Skip Steps` / `Native Scan`	`Held Press`	What **holding** Fast Forward or Rewind does. A single tap is not affected. See Fast Forward Mode.
Fast Forward Step Interval	Number, 100–2000 ms	`250`	The gap between repeats while Fast Forward or Rewind is held. Used only by **Remote Presses** and **Skip Steps**. See Fast Forward Step Interval.
Deep Link Region	Text	*(blank)*	Two-letter ISO 3166-1 country code, e.g. `IN`, `US`, `GB`. See Deep Link Region.

Support and Troubleshooting

PROPERTY	TYPE	DEFAULT	WHAT IT IS
Control Port	Number, 1–65535	`49153`	The port the driver connects to. Every Apple TV seen so far uses `49153`, so the default is usually already right. **Find Apple TVs reads the real port off the device** and fills it in, so you do not have to rely on that. Type it by hand only if discovery cannot run on your network. Changing it disconnects and redials.
Log Level	`0 – Fatal` … `5 – Trace`	`2 – Warning`	How much the driver writes when logging is on.
Log Mode	`Off` / `Print` / `Log` / `Print and Log`	`Off`	Where logging is written. **`Off` means nothing is written at any level.** Set it to `Print and Log` before reproducing a problem, and put it back afterwards — logging turns itself off after 45 minutes anyway.

The driver ships silent on purpose. Both properties have to move to get output: raising Log Level alone does nothing while Log Mode is `Off`.

Actions

Eight actions, on the driver's Actions tab. Everything a dealer needs is here; everything that existed only for bench work has been removed.

ACTION	PARAMETER	WHAT IT DOES
Find Apple TVs	—	Looks for Apple TVs on the network and fills in **Discovered Apple TVs**. Takes a few seconds. Finding nothing is almost always the network rather than the Apple TV — see Pairing.
Pair with Apple TV	—	Starts pairing. The television shows a four-digit code. Refuses, with a reason in Last Error, if there is no address or if the Apple TV has asked for a wait.
Enter Pairing Code	`Pairing Code` (text)	The second half of pairing. Four digits, nothing else. On success the credential is stored, Pairing Status reads `Paired`, and the driver connects.

ACTION	PARAMETER	WHAT IT DOES
Clear Pairing	—	Forgets the stored credential and drops the session. Also the manual way out of a pairing somebody started and walked away from. **There is no undo** — the only way back is to pair again at the device.
Connect Now	—	Connects immediately, ignoring the reconnect back-off. This is the button to press after typing the IP address.
Disconnect	—	Drops the session and stays down until something reconnects it. Useful when moving the Apple TV or changing its network.
Refresh Status	—	Asks the Apple TV what it is doing and updates **Power State**. The only on-demand way to refresh that field. Every outcome is reported in **Last Error**: on some newer tvOS builds the Apple TV refuses the request, and Power State then keeps whatever the device last pushed; with no session it says so and points you at **Connect Now**.
List Apps	—	Writes every installed app and its bundle identifier to the Director log, so you can copy an id into **App Deep Links**. It raises the log for its own duration and puts your Log Mode and Log Level back afterwards, so it works with logging off.

Programming commands

Nine commands, available on the driver in the Programming tab.

COMMAND	PARAMETER	NOTES
Send Remote Key	`Key` (list)	The remote buttons. See the list below.
Send Media Command	`Command` (list)	The transport commands. See the list below.

COMMAND	PARAMETER	NOTES
Launch App	`Bundle Identifier` (text)	Opens an app at its home screen, e.g. `com.netflix.Netflix`. Get ids from **List Apps** or **Installed Apps**.
Launch Deep Link	`Deep Link URL` (text)	Opens a URL. This is the command for "open *this title*" rather than "open this app". See Deep links.
Volume Up	—	One volume-up press on the Apple TV.
Volume Down	—	One volume-down press.
Next Track	—	Skip forward one track/chapter.
Connect	—	Connects immediately, ignoring the reconnect back-off.
Disconnect	—	Drops the session.

Send Remote Key accepts: `up`, `down`, `left`, `right`, `select`, `menu`, `home`, `volume_up`, `volume_down`, `play_pause`, `channel_up`, `channel_down`, `guide`, `page_up`, `page_down`, `siri`, `screensaver`, `sleep`, `wake`.

Send Media Command accepts: `play`, `pause`, `next`, `previous`, `skip_forward`, `skip_backward`, `stop`, `fast_forward_begin`, `fast_forward_end`, `rewind_begin`, `rewind_end`.

> `fast_forward_begin` / `fast_forward_end` and `rewind_begin` / `rewind_end` are a **held** scan: send the `begin`, then send the matching `end` when the button is released. Sending a `begin` without its `end` leaves the Apple TV scanning.

Anything sent while the session is down is refused with a sentence in Last Error, and the driver starts reconnecting.

Apps, deep links and the App Switcher

How a mini-app tile launches an app

When a customer presses a mini-app tile bound to **App 1 … App 25**, the driver works out which Apple TV app that slot means, in this order:

1. A line in **App Deep Links** for that slot number.
2. A line in **App Deep Links** naming that mini-app driver.

3. An **exact** name match against the app list read from the Apple TV — or the same app under another name it is known by. tvOS lists Apple's own app as *TV*, so a mini-app called **Apple TV** finds it; **HBO Max** and **Max** are one app, so a mini-app called either finds the Apple TV's listing of it, whichever of the two names that Apple TV uses. This is not a guess and is not logged as one.
4. A **close** name match against that list. This one is logged as approximate. If two installed apps match the name equally well the driver **refuses and names both** rather than guessing — add a deep-link line to settle it.
5. The mini-app driver's own `APP_ID` variable, if it looks like an Apple bundle identifier.

When rule 3 or 4 wins, the driver **writes the answer back into App Deep Links** as a slot line. Your own text is preserved byte for byte and the new line is appended — the driver never rewrites, reorders or deletes a line you wrote. The next press then takes rule 1 and never guesses again. Edit or delete that line to change what the slot launches.

A slot that will not resolve re-reads the app list **once** before refusing.

Choosing the same tile again within three seconds of a launch does not launch the app a second time, because one choice may arrive twice. A press that was refused does not count, and with **Passthrough Mode** `Off` the room is still switched onto this driver.

App Deep Links format

One override per line. Two forms:

```
3101=com.netflix.Netflix
Netflix=com.netflix.Netflix
```

- `<slot>=<bundle id>` — a slot line. Wins over everything else.
- `<name>=<bundle id>` — a name line, where `<name>` is the **bound mini-app driver's own name**. A name line survives re-binding that mini-app to a different slot; a slot line does not.
- A key that is entirely digits is read as a slot; anything else is read as a name. So `9Now=com.example.app` is a name line, not slot 9.

Slots are `3101` (App 1) through `3125` (App 25).

Installed Apps and Selected App Id

Installed Apps is filled from the Apple TV each time the driver connects. Pick a row and **Selected App Id** shows that app's bundle identifier, ready to copy into **App Deep Links** as `<slot>=<id>`. Picking a row does not launch anything. Picking the blank first row clears the id.

The Apple TV sends **no notification** when an app is installed or removed, so the list is refreshed on connect, when another driver asks for it and the driver's copy is more than 15 minutes old, when the **List Apps** action runs, and once when a slot will not resolve. There is deliberately no polling.

Deep links: opening a title

Launch Deep Link takes a URL. Paste an ordinary web address and the driver rewrites it into the form the app on the Apple TV actually opens. Four rules, first match wins:

IF THE URL CONTAINS	THE DRIVER SENDS
`youtube`	the same URL with `https` replaced by `youtube`
`primevideo.com`	a Prime Video link built from the URL's `gti=` value and your **Deep Link Region**
`netflix`	a Netflix link built from the digits after `/title/` or `/watch/`
`hotstar`	a Hotstar link built from the last path segment, with any `?query` or `#fragment` cut off first — so a share link carrying `?utm_source=…` works

Anything that is not an `http://` or `https://` address — a bundle identifier, or a URL that already carries an app's own scheme — is passed through untouched.

Two things worth knowing before you write programming:

- **An ordinary `https://` Netflix link opens the Netflix app on its home screen, not the title.** That is why the rewrite exists. The web URL was measured twice on real hardware and opened the app both times. That the rewritten form starts the title was measured once, on 2026-08-30, through an action that no longer ships — the rewriter is pinned to that same string by the driver's own fixtures, so the two agree by construction rather than by a second bench run.
- **When a rule cannot do its job it says why and sends your link unchanged** rather than mangling it — a Netflix URL carrying a slug instead of digits, a Prime URL with no `gti=`, or a Prime URL when no region is known. The unchanged link still opens the app; it just opens it at the home screen instead of the title. The reason is the part you can fix.

Paste carefully. If a target arrives wrapped in quotes, angle brackets or markdown link syntax, the driver strips the wrapping and logs both the string you gave it and the string it actually sent. If you see that pair in the log, something in your copy path changed the link — fix the paste, not the driver.

"Sent" is not "opened." A launch the Apple TV refuses outright comes back with the Apple TV's own error in **Last Error**. A launch it accepts is not a promise that the app opened the *content* — nothing reports that back. So if nothing was refused and the television did nothing, the address you typed is what to change.

Passthrough Mode

What happens to the **room** after a mini-app tile launches an app.

VALUE	BEHAVIOUR
`Off` *(default)*	The room is switched onto this driver, so the remote's buttons, the transport controls and the volume drive the Apple TV directly.
`On`	The mini-app stays selected in the room and passes its buttons on to the Apple TV through this driver — **but only while the room has an audio path for it** (below).

`On` needs an audio path. `Off` does not. Control4's mini-app drivers pass a button on only to a device that is on the room's current audio path. Their own documentation says a valid audio route for the mini-app "is required for passthrough mode to work", and that without one "you will not be able to control the media player/TV". So before you choose `On`, open **System Design**, select the room, and check that **Audio Path** reads **YES** for the **App Switcher** and for every mini-app bound to it.

If it reads **NO**, pressing a tile still opens the app, and then the mini-app's buttons do nothing. Nothing reaches this driver, so nothing is logged here and **Last Error** stays empty. Set **Passthrough Mode** back to `Off`.

Before version 1091 this driver did not declare the setting that lets the Apple TV carry the App Switcher's audio, and a site running 1087 saw **Audio Path = NO** for the App Switcher and every mini-app. Version 1091 declares it, which Control4's SDK says this layout needs. If **Audio Path** still reads **NO** after updating, use `Off` and tell DTI what System Design shows.

Deep Link Region

A two-letter ISO 3166-1 country code, e.g. `IN`, `US`, `GB`.

Only Prime Video uses it. Prime deep links carry the region as a `territory=` value and will not open a title without one; YouTube, Netflix and Hotstar links ignore this field entirely.

Leave it blank to use the country code Director's own geolocation reports. Anything typed here **wins** over that — which is what a project behind a VPN, or one whose controller geolocates to the wrong country, needs. Set it before using **Launch Deep Link** with a Prime Video address.

Fast Forward, Rewind and subtitles

Holding Fast Forward or Rewind

- **A tap** jumps 10 seconds forward or back, whatever Fast Forward Mode is set to.
- **Holding** the button scans. With the default **Held Press** mode it works like the Apple TV's own Siri Remote: the picture keeps moving while the button is held, and plays from where you let go.

Two things are normal and not a fault:

- **Each hold starts with a 10-second jump.** The Control4 app sends a normal tap the instant the button goes down, and only starts the hold about half a second later. The scan follows the jump.
- **It depends on the app that is playing.** Fast Forward is handled by the app on screen, so the Apple TV app, Netflix and YouTube can each behave a little differently. Apple's own remote behaves differently between apps in the same way.

A hold always ends when the button is released. If the Control4 app never sends the release — the app is closed mid-press, or the network drops — the driver stops on its own: after 30 seconds with **Held Press**, after 40 repeats with **Remote Presses** and **Skip Steps** (10 seconds at the default interval), and after 6 seconds with **Native Scan**.

Fast Forward Mode

Composer property, **Apps and App Switcher** section. It only changes what **holding** the button does.

SETTING	WHAT HOLDING FAST FORWARD DOES	WHEN TO USE IT
Held Press *(default)*	Holds the button down until you let go, as the Siri Remote does.	The recommended setting for most apps.
Remote Presses	Repeats a short press, giving a steady run of 10-second jumps — exactly what holding the d-pad right or left does.	If an app does not scan with Held Press.
Skip Steps	Repeats a jump that grows the longer you hold: about 40 seconds for one second of holding, about 2 minutes 40 seconds for three.	If an app ignores both of the above. Some apps act only on the first jump of a run.
Native Scan	Asks the Apple TV to start and stop its own scan.	Rarely useful; most apps do not respond to it.

Rewind follows the same setting, in the other direction.

Changing it needs no reboot or reconnect — the next press uses the new setting. If you upgraded from an older version of this driver, Composer shows the default the first time; nothing needs setting.

Fast Forward Step Interval

Composer property, **Apps and App Switcher** section. Milliseconds between repeats while the button is held, from 100 to 2000. Default `250`.

- Used only by **Remote Presses** and **Skip Steps**. **Held Press** and **Native Scan** ignore it.

- **If holding moves the picture only once**, the app is probably still seeking when the next repeat arrives. Raise the value — try `600`, then `1000`.
- **Lower is faster.** `150` matches the speed of holding the d-pad.
- A value outside 100–2000 is ignored and `250` is used.

Subtitles (CC)

The Apple TV remote in the Control4 app has a **CC** button. Press it to switch subtitles **on or off**; press it again to switch them back.

- **The driver checks the change actually happened.** It reads the Apple TV's subtitle setting, switches it, waits a moment and reads it again. If the setting did not change, **Last Error** says so — the button never fails silently.
- **It depends on the app.** Subtitle support belongs to the app that is playing. If an app does not allow subtitles to be switched this way, use the Apple TV's own subtitle menu: swipe down on the Siri Remote during playback.
- **CC switches subtitles on and off only.** Choosing a subtitle language is done in the Apple TV's on-screen menu.

The remote in the Control4 app

The driver tells the Control4 app which remote buttons it supports: Play, Pause, Stop, Fast Forward, Rewind, Skip Forward and Skip Back, plus Home, Back, Guide and **CC**.

If the remote looks unchanged straight after updating the driver, close and reopen the Control4 app.

Guide opens the app switcher. The Guide button, on a Control4 remote or in the Control4 app, presses the Apple TV's TV button twice. That opens the screen of open apps, where you swipe an app up to close it. To send the Apple TV's own Guide key instead, which does something only in a live-TV app, set the **Guide Button** property to **Guide**.

In the app switcher, each press of Up closes one app. Left and Right move between apps, Select opens one, and Up closes the one in the middle. Holding Up still closes only one. Back, Home or Select leave the app switcher, and Up is the ordinary Up button again.

Play only resumes what is paused

The **Play** button resumes something that is paused on the Apple TV, and does nothing when nothing is. Pressing Play with the Apple TV on its home screen, on its screensaver, or in an app that is not playing anything does not start Apple Music or anything else.

With a film or song paused, Play presses the Apple TV's **Play/Pause** button first. The button clears a screensaver, even one the Apple TV did not report. Without a screensaver it simply starts what was paused: the Apple TV says it is playing, and nothing more is sent. Only when it does not say so, because the button was used up clearing the screensaver, does Play follow.

Play on something already playing, and Pause on something already paused, send nothing. Some apps take Play and Pause as a toggle, so sending them anyway paused a playing film or started a paused one. The log says `Apple TV: Play was not sent — the Apple TV says it is already playing` or `Apple TV: Pause was not sent — the Apple TV`

`says it is already paused`. A Pause pressed during the two-second wait described below also sends nothing, because the Apple TV has already said it is paused.

Press Play again within three seconds to send it anyway. An app can pause without telling the Apple TV, and the driver then still believes it is playing. A second Play straight after one that was not sent goes out as a plain Play. On an app that takes Play as a toggle, that second press pauses something that really was playing.

A pause shorter than two seconds does not fire the Pause event. Every skip makes the Apple TV pause for a moment while it seeks, then play again. So when something that was playing pauses, the driver waits two seconds before it tells Control4. If it is playing again by then, neither the Pause nor the Play event fires, and programming on them does not run once per skip. A real pause fires its Pause event two seconds late. Play and Stop events are never held.

The driver only resumes what the Apple TV has told it is playing or paused since the driver connected. Straight after the driver connects the Apple TV has told it nothing, even when an app such as Apple Music is sitting there ready, so Play does nothing until something is played or paused on the Apple TV. The driver's log says so: `Apple TV: Play was not sent – the Apple TV has not said that anything is paused`. Play stops working again when the Apple TV says the player has gone: the film ended, the app was closed, or the Apple TV went to sleep. The log then says `Apple TV: Play was not sent – nothing is paused on the Apple TV`.

Leaving an app forgets what was paused, whichever remote did it. After **Home** on the Control4 remote or the Siri Remote, after opening another app, or after **Info** or an app launched from Control4, Play does nothing until something is played or paused on the Apple TV again. So music paused in Apple Music does not start when someone presses Play on the Home screen.

The Apple TV tells the driver whenever the app in front changes, but not which app it is, so going back into the same app counts as a change too. Back in the app, start it once on the Apple TV, and Play and Pause from Control4 work as before. The screensaver does not count: a film paused under it still resumes.

This applies to the **Play** button on the remote and to the Play command in programming. **Send Media Command** with `play`, and **Send Remote Key** with `play_pause`, are not held back: they send exactly what you choose. Every other button is unchanged.

Search

Apple TV Search appears in the Control4 app next to the Apple TV. It connects itself to the Apple TV when the driver is added: on the **Connections** tab, its **Search Output** is joined to the Apple TV's **Search Input** without anything being dragged.

Adding this to a project that already has the Apple TV driver: remove the driver and add it again. Updating the driver in place does not create the **Apple TV Search** tile, and does not connect it to the Apple TV. An **Apple TV Keyboard** left over from an earlier version is no longer part of the driver and does nothing: remove it from the project.

Apple TV Search

A search box with a remote under it, so you can type and then pick the result on the television without leaving the page.

1. Select **Apple TV Search** in the Control4 app. The room switches to the Apple TV, and the Apple TV opens its own Search app on the television.
2. Type what you are looking for. The text appears in the Apple TV's search box a moment after you stop typing, and the Apple TV shows its results.
3. Use the arrows to move to the result you want, and **OK** to open it.

CONTROL	WHAT IT DOES
Typing	Sends the whole box to the Apple TV half a second after you stop.
Search / **Enter** on the keyboard	Sends it straight away and closes the keyboard, so the remote is in view.
Arrows and **OK**	Move around the television's screen and open what is highlighted, like the Siri Remote.
Delete (top left)	Deletes the last character - from the box on the phone and from the Apple TV's search box.
Clear (top right)	Empties both boxes.
Back (bottom left)	Goes back, like the Siri Remote's back button.
Home (bottom right)	Goes to the Apple TV's home screen.

Good to know:

- A message under the box says why when something could not be sent - for example, the Apple TV is still connecting.
- Once the arrows have moved off the search box onto the results, the Apple TV has no text box to type into. If the page says so, move back to the search box on the television (or close and reopen **Apple TV Search**) and type again.
- The page works from the Control4 app on phones and tablets, and on Control4 touchscreens.

Programming with the Apple TV's own state

The Apple TV reports what its screen is doing, and this driver publishes it three ways so Control4 can act on it. **All three move when somebody uses the Apple Remote** — nothing has to go through Control4 for them to fire.

Events, on the driver's Events tab in Composer programming:

EVENT	FIRES WHEN
App In Use	the Apple TV shows an app rather than the home screen
Home Screen Shown	it returns to the home screen
Screensaver Started	the screensaver starts
Went To Sleep	it goes to sleep
Screen State Changed	any of the above, for programming that re-reads the variable

A variable, `SCREEN_STATE`, holding `asleep`, `screensaver`, `idle` or `awake`. Use it in a conditional — *if SCREEN_STATE is awake*.

The Screen State property, so a dealer can watch it move on the Properties tab without writing any programming at all.

On tvOS 26.6 the home screen reads `awake`. The Apple TV seen on tvOS 26.6 never reported its home screen as `idle`, so **Home Screen Shown** did not fire, and **App In Use** fired when the screensaver ended, whatever was on screen. **Screensaver Started** and **Went To Sleep** fired as described.

A worked example — lights down when the television starts its screensaver, and back up when someone starts watching again:

```
When  Apple TV -> Screensaver Started
Then  Set Living Room lights to 20%

When  Apple TV -> App In Use
Then  Set Living Room lights to 70%
```

Play, Pause and Stop

The Apple TV's standard Media Player events, **Play**, **Pause** and **Stop**, follow what the Apple TV reports it is doing, whichever remote was used: the Siri Remote, the Control4 app, or an iPhone. Each fires **once per change**, so programming on **Play** runs when playback starts and does not

run again while it continues. **Stop** also fires when the Apple TV goes to sleep with something playing or paused. The driver's log shows one line per change, such as `Apple TV: reporting 'paused' to the Media Player proxy`.

```
When  Apple TV -> Play
Then  Set Living Room lights to 10%

When  Apple TV -> Pause
Then  Set Living Room lights to 40%

When  Apple TV -> Stop
Then  Set Living Room lights to 70%
```

What this does not tell you is which app. The Apple TV reports the state of its screen, not the name of the foreground application, and nothing in the protocol this driver speaks carries the foreground app's name. **Last Launched App** is the nearest thing, and it is what this driver last *asked* the Apple TV to open rather than what is on screen now. If you need the running app's name, ask DTI.

Programming examples

These are the shapes a dealer actually writes in the Programming tab.

Watch a film in the cinema room (button or scene)

```
When Cinema Keypad button 1 is pressed:
    Set Room 'Cinema' source to 'Apple TV'
    Delay 00:00:02
    Apple TV -> Launch Deep Link
        Deep Link URL: https://www.netflix.com/title/83067197
```

The delay gives the room time to switch video before the Apple TV is asked to open anything.

Put an app on a keypad button

```
When Media Keypad button 3 is pressed:
    Apple TV -> Launch App
        Bundle Identifier: com.netflix.Netflix
```

A "goodnight" macro that puts the Apple TV to sleep

```
When 'Goodnight' is executed:
    Apple TV -> Send Remote Key
        Key: sleep
```

Wake the Apple TV when the room turns on

```
When Room 'Media' is turned on:
    Apple TV -> Send Remote Key
        Key: wake
```

Free the socket while the house is empty, and take it back on arrival

```
When 'Away' is executed:
    Apple TV -> Disconnect

When 'Home' is executed:
    Apple TV -> Connect
```

Transport on a third-party remote

```
When <remote> FFWD is pressed:
    Apple TV -> Send Media Command
        Command: fast_forward_begin

When <remote> FFWD is released:
    Apple TV -> Send Media Command
        Command: fast_forward_end
```

React to the Apple TV going to sleep

The **Power State** property changes to `off` when the Apple TV reports itself asleep, so it can drive programming through a property-changed event on the driver.

Troubleshooting

Read Last Error first. It needs no logging at all, it carries the reason in the words of whatever refused, and it is blank whenever nothing has failed.

On a DriverCentral build, rule out the licence before anything else in this section. An unlicensed driver refuses every action and command in this manual and writes one sentence for whichever was refused, naming it:

> PRX_CMD.PLAY was refused because the DriverCentral licence for this driver is invalid or expired …

The entry name changes with what you pressed; the rest does not. While that sentence keeps coming back, nothing below applies — the driver is not failing, it is unlicensed. See Licensing and support.

Every sentence in the left-hand column below is a sentence this driver actually writes. They are abbreviated with … where they are long; match on the opening words.

Nothing connects

The driver keeps trying by itself. A connection attempt that gets no answer is given up on after 30 seconds, hung up, and retried; after two failures in a row the driver searches the network for the Apple TV it is paired to and follows it if its address or its port has changed. It recognises that Apple TV by the identity it announces on the network, which the driver remembers when you pick the Apple TV and reads again after it connects — when the controller can hear those announcements. Where it cannot, the driver leaves the address alone and says so in the log, and a DHCP reservation is what keeps the address still. If you are watching Connection Status, expect it to alternate between **Connecting** and **Offline** while a search is running rather than sitting on one of them.

It never overrides an address you typed. If you change **Apple TV IP Address** by hand, the driver does not go looking for the Apple TV elsewhere until it has connected at the address you set, and from then on it follows the Apple TV that answered there — a typed address can be a different Apple TV. **Clear Pairing** and a new pairing do the same: the driver forgets which Apple TV it was following until it has connected again.

LAST ERROR SAYS	WHAT IT MEANS	WHAT TO DO
`no Apple TV IP Address is set, so there is nothing to connect to`	The address field is empty.	Type the Apple TV's IPv4 address, then run **Connect Now**.
`no Apple TV IP Address is set – enter the address in Composer, then run Connect Now`	Same, reported at startup.	As above.

LAST ERROR SAYS	WHAT IT MEANS	WHAT TO DO
`this driver has no pairing credential – run the Pair with Apple TV action, and type the four-digit code the television puts on screen`	Normal first-run state.	Pair. See Pairing.
`could not create the network connection: …`	Director would not open the socket.	Check the address and port, and that the controller can reach the Apple TV's subnet.
`pairing is still running, so this driver will not start a session yet – type the four-digit code the television is showing, or run Clear Pairing to abandon it`	A pairing was started and never finished.	Type the code, or run **Clear Pairing**.
`The Apple TV this driver is paired to did not answer, so its address was left alone. It may be powered off, or on a different network from the controller.`	The driver searched the network for its Apple TV and nothing announcing that identity answered.	Check the Apple TV is on and awake, and on the same network as the controller. If it has been replaced, pick the new one in **Discovered Apple TVs** and pair it.
`The Apple TV moved from <old address> to <new address> and this driver followed it. Nothing needs re-pairing.`	The Apple TV came back on a different IP address and the driver followed it.	Nothing. A DHCP reservation stops it happening again: reserve the address **Apple TV MAC** shows once it matches the one under **Settings** → **General** → **About** on the Apple TV.

LAST ERROR SAYS	WHAT IT MEANS	WHAT TO DO
`The Apple TV is still at <address> but is now answering on port <port> instead of <old port>, and this driver followed it. Nothing needs re-pairing.`	The Apple TV restarted and chose a new port, as it may.	Nothing.
`Composer shows driver version <N>, but the code running on the controller is from version <M>, so the driver behaves as version <M>. Install a version higher than both, or remove the driver and add it again.`	The controller is running the code of a different build from the one whose settings Composer shows — seen after a lower version was installed over a higher one of the same file. Settings that exist only in the version Composer shows may do nothing (a DTI key typed there is never checked, for example).	Install a version higher than both numbers, or remove the driver and add it again. Never install a lower version over a higher one.

The Apple TV stays asleep after Room On. An Apple TV that goes to sleep leaves the network within about a minute, and nothing on the network can then reach it to wake it. So when Room On finds it not answering, the driver sends it a **Wake-on-LAN** packet, to the address **Apple TV MAC** shows, and keeps trying for three minutes. The moment it answers, the driver wakes it. Wake-on-LAN can only work for an Apple TV on **Ethernet**, never over Wi-Fi, and many Apple TVs ignore it in deep sleep. The log says when one was sent: `Apple TV: sent Wake-on-LAN to the Apple TV (Room On)`. If it says `Apple TV: no Wake-on-LAN was sent – the driver does not know the Apple TV's hardware address yet`, **Apple TV MAC** is still blank: it fills in once the Apple TV has connected.

The better fix is to stop it sleeping. Set **Send Power OFF/Sleep** to **No**, set **Sleep After** to **Never**, and on the Apple TV turn off **Settings** → **Remotes and Devices** → **Control TVs and Receivers**. With that setting on, the Apple TV can go to sleep when Room Off turns the television off, because the television tells it to over HDMI.

The Apple TV goes to sleep at Room Off with Send Power OFF/Sleep at No. This driver did not do it: at **No** it sends the Apple TV nothing at Room Off, and its log says `Apple TV: Room Off did not put the Apple TV to sleep – Send Power OFF/Sleep is No`. Look for whatever else acts on Room Off:

- another driver for the same Apple TV: a second copy of this driver (its **Send Power OFF/Sleep** starts at **Yes**), an infrared Apple TV driver, or another maker's Apple TV driver;
- programming that sends **Send Remote Key** with `sleep`;
- the television, receiver or HDMI switch, over HDMI, as above.

The log shows when it happened: `Apple TV: the screen is now 'asleep'`. A few seconds after Room Off means something acted on Room Off; many minutes later is the Apple TV's own **Sleep After**.

Pairing

LAST ERROR SAYS	WHAT IT MEANS	WHAT TO DO
`the Apple TV asked for a short wait after the last pairing attempt and N second(s) of it are left…`	You pressed Pair inside the device's own back-off.	Wait the stated number of seconds, then press Pair **once**.
`the code typed is not a pairing code: the television shows FOUR DIGITS, and nothing else is accepted`	A letter, symbol or over-long code was typed, or none at all. What was typed is not repeated here, because Pairing Code is masked like a password.	Re-read the code off the television.
`the pairing code was not accepted: …`	The Apple TV rejected the code.	Run **Pair with Apple TV** again and read the new code off the screen.
`pairing produced no credential`	The exchange completed without producing anything usable.	Run **Clear Pairing**, then pair again.
`pairing could not be started: …`	The attempt failed before it began.	Check the address and port; run **Connect Now**, then pair.

Buttons and commands do nothing

LAST ERROR SAYS	WHAT IT MEANS	WHAT TO DO
`the Apple TV session is not up yet, so '<command>' was not sent`	The command arrived before the session was ready. The driver is already reconnecting.	Wait, or press **Connect Now**. If it persists, check Connection Status.

LAST ERROR SAYS	WHAT IT MEANS	WHAT TO DO
`the Apple TV session is not up yet, so '<target>' was not launched`	Same, for a launch.	As above.
`this driver has no mapping for '<command>', so the button did nothing`	A proxy command this driver does not implement.	Nothing to fix on this driver; the button has no Apple TV equivalent.
`'<key>' is not a button this driver can send`	A **Send Remote Key** value outside the list.	Pick from the list in Programming commands.
`'<command>' is not a media command this driver can send`	A **Send Media Command** value outside the list.	As above.
`the Apple TV accepted the subtitle change but its setting did not change…`	CC switches subtitles and then checks the Apple TV really changed. This app acknowledged the switch but kept its setting.	Use the swipe-down overlay on the Apple TV for this app, and tell DTI which app it was.
`the Apple TV reported a subtitle setting this driver does not recognise…`	The Apple TV reported a subtitle state this driver has not seen before, so it changed nothing rather than guess.	Use the swipe-down overlay, and send DTI the Dev log from the press.
`'<char>' was not sent: this driver has no digit or symbol keys…`	Expected. Digits and symbols are not remote keys.	Use the **Search** tab for text.
`<name> was not sent: <reason>`	A proxy command with no Apple TV equivalent — no disc tray, no tuner, no DVR, no mute, no aspect-ratio control.	Expected; the reason names why.

Holding Fast Forward or Rewind does nothing, or moves the picture only once. These produce no Last Error, because the Apple TV accepted the command — the app on screen chose not to act on it. Work down Fast Forward Mode: **Held Press**, then **Remote Presses**, then **Skip Steps**, and raise Fast Forward Step Interval if the picture moves once and stops. Note which app it was when reporting it.

Play does nothing. Look in the driver's log for `Apple TV: Play was not sent – …`. `nothing is paused on the Apple TV` means the Apple TV has said there is no film or song to resume: it ended, its app was closed, the Apple TV went to sleep, or the app in front changed (Home, or another app). That is intended: Play no longer starts or resumes Apple Music from the home screen. `the Apple TV has not said that anything is paused` means the driver has only just connected and the Apple TV has not reported anything yet; start or pause something once from the Apple TV, and Play from Control4 works after that. If something really is paused and Play is still held back, send DTI the Dev log from the press.

Connection Status shows Offline for a few seconds now and then. Some Apple TVs (seen on tvOS 26.6) stop answering on the driver's remote session while the network connection stays up. While that lasts the driver hears nothing: no pause, no screensaver. The driver notices within about 16 seconds and reconnects by itself, keeping track of what was paused. The log says `Apple TV: the Apple TV stopped answering on its remote session, so the session is being reopened`.

The CC button is missing from the Control4 app. Close and reopen the app after updating the driver. If it is still missing, check the driver really updated: **Driver Version** on the driver's properties should show the new version.

App tiles and launching

LAST ERROR SAYS	WHAT IT MEANS	WHAT TO DO
`slot N is bound to '<app>', and none of the M app(s) installed on this Apple TV matches that name. Add a line 'N=<bundle id>' to the App Deep Links property – run the List Apps action to see every id.`	The mini-app's name does not resemble anything installed. **`0 app(s)` and `18 app(s)` mean different things** — zero means the list was never read.	If `0`: connect the driver first. Otherwise check the app is installed on the Apple TV (**Installed Apps**), then add the deep-link line. A name line such as `HBO Max (International)=<id from List Apps>` survives re-binding the mini-app.
`slot N is bound to '<app>', and this Apple TV's app list has not been read yet…`	No session yet.	**Connect Now**, then press the tile again.

LAST ERROR SAYS	WHAT IT MEANS	WHAT TO DO
`slot N is outside this driver's app range (3101–3125)`	A binding outside App 1–25.	Re-bind the mini-app to a real slot.
`SET_INPUT carried no usable slot number`	Malformed input request.	Check the mini-app binding.
`SET_INPUT (N) was not sent: it arrived on binding B, not the App Switcher proxy (5999); only the App Switcher owns app slots`	A wiring problem — an input request reached the wrong proxy.	Bind mini-apps to the **App Switcher**, not to the main proxy.
`the mini-app on slot N has no APP_NAME variable…`	The bound driver exposes no name to match on.	Add `N=<bundle id>` to **App Deep Links**.
`Launch App was given nothing to launch`	Empty parameter.	Fill in the bundle identifier.
`what Launch App was given was nothing but wrapping – quotes, angle brackets or markdown syntax with no link inside them`	The pasted value contained no link.	Re-copy the link from an address bar.
`Launch Deep Link was given no URL`	Empty parameter.	Fill in the URL.
`'<row>' is not in the app list this driver last read from the Apple TV (N app(s))…`	The **Installed Apps** selection is stale.	Connect the driver and open the property again to refresh the list.
`error 4 from FBSOpenApplicationErrorDomain` *(and similar)*	The Apple TV itself refused the launch and this is **its** error, passed	Usually the app is not installed, or the link is not one that app accepts.

LAST ERROR SAYS	WHAT IT MEANS	WHAT TO DO
	through unchanged.	

A mini-app opens its app, and then its buttons do nothing. There is no Last Error for this, because the presses never reach this driver. It happens with **Passthrough Mode** set to `On` when the room has no audio path for the mini-app. Check **Audio Path** in System Design as described under Passthrough Mode, or set **Passthrough Mode** to `Off`.

Volume

LAST ERROR SAYS	WHAT IT MEANS	WHAT TO DO
`SET_VOLUME_LEVEL arrived with LEVEL = '<x>', which is not a number, so nothing was sent`	Bad programming argument.	Send a number.
`the Apple TV refused an absolute volume level…`	This Apple TV does not accept a level. The driver falls back to repeated volume presses and says so once per session.	Nothing to fix; up/down still work.

Getting a log for a support ticket

1. **Log Mode** → `Print and Log`.
2. **Log Level** → `4 - Debug` or `5 - Trace`.
3. Reproduce the problem.
4. Copy the Director log.
5. Put **Log Mode** back to `Off` (the driver does it for you after 45 minutes).

At the shipped defaults the driver writes **nothing at any level**, which is deliberate. Both properties have to move.

A log that covers the driver starting (after an update, say), or a change to **Log Level** or **Log Mode**, carries one line naming the build that wrote it. On a licensed build it reads `Apple TV: driver version <N>, release build`, and on a bench build DTI gave you for testing it reads `Apple TV: driver version <N>, bench build - development detail is logged`, where `<N>` is the build number **Driver Version** shows. If you set the two properties before an update, that line is in your capture.

Include in any report: the **Driver Version** value, the controller model and its OS version, the Apple TV's model, and the exact **Last Error** text.

Licensing and support

This driver carries **two** licence systems and **either one satisfies it**:

- **DriverCentral**, for dealers who bought it on drivercentral.io. Their licensing client adds **Cloud Status** and **Automatic Updates**.
- **DTI activation**, for dealers who bought it directly from DTI Automation. A key is typed into **Activation Key** and the driver validates it itself.

You need only the one that matches where you bought the driver. Both appear on the Properties tab in every build because it is one driver file, not two — an entry refused by one system is allowed the moment the *other* says yes, so a DriverCentral customer never needs a DTI key and a DTI customer never needs a DriverCentral account. See Cloud Settings.

While either licence is valid, the driver behaves exactly as documented here.

If the licence is invalid or expired, the driver does nothing. Every command, action and request it answers is refused until the licence is activated — not only the remote keys, media commands, launches, app-slot selection, volume and search you pay for, but **Pair with Apple TV**, **Enter Pairing Code**, **Clear Pairing**, **Connect Now**, **Disconnect**, **Refresh Status** and **List Apps** as well. The driver stays loaded and names the entry it refused, putting this sentence on **Last Error**:

> the DriverCentral licence for this driver is invalid or expired (Cloud Status in Composer shows the account state). Renew or activate it on drivercentral.io and the command works again immediately - no restart needed.

A DTI-licensed install that has not been activated is refused the same way and for the same reasons; its refusal names DTI rather than DriverCentral, and **Activation Status** rather than Cloud Status is where the reason is read.

You do not have to press anything to find this out. An unlicensed driver says so on **Last Error** as soon as it loads, and keeps saying it — the sentence is re-asserted rather than cleared, so it cannot be wiped out by the driver going about its business. Saying why it will not work is the only thing an unlicensed install does. Read **Cloud Status** on the Properties tab for the account state itself.

Licence state is checked live — renewing on drivercentral.io restores every command immediately, with no driver reload and no controller restart.

Before you contact support

Have these ready. They answer most of the first round of questions:

1. **Driver Version**, exactly as the property reads it.
2. **Pairing Status**, **Connection Status** and **Last Error**, copied out.
3. Controller model and OS version; Apple TV model.
4. What you did, and what happened instead.
5. A log, if the problem is reproducible — see Getting a log for a support ticket.

Apple TV is a trademark of Apple Inc. This driver is not made by, endorsed by or affiliated with Apple Inc.