

Overview

> DISCLAIMER: This software is neither affiliated with nor endorsed by either Control4 or MQTT.

The MQTT Universal driver brings MQTT devices and values into Control4. You add relays, contact sensors, buttons, events and variables, and give each its own state and command topics. It reaches the broker through the MQTT Broker driver.

Index

- System Requirements
- Features
- Installer Setup
  - Driver Installation
  - Driver Setup
    - Driver Properties
      - Cloud Settings
      - Driver Settings
      - Availability Settings
      - Add Devices
      - Add Variables
      - Manage Items
      - Item Configuration
  - Connections
- Programming
  - Events
  - Conditionals
  - Commands
  - Variables
- Developer Information
- Support
- Changelog

System Requirements

- Control4 OS 3.3+
- MQTT Broker driver installed and connected

Features

Devices:

- **Relays** - Create `RELAY` bindings for on/off control with state feedback
- **Contact Sensors** - Create `CONTACT_SENSOR` bindings for open/closed state
- **Buttons** - Publish press commands to MQTT topics, with optional keypad linking
- **Events** - Create `BUTTON_LINK` bindings that trigger on MQTT messages, for keypad integration

Variables:

- **String/Bool/Number/Float** - Create Control4 variables synced with MQTT
- **Temperature** - Create `TEMPERATURE_VALUE` bindings for thermostat integration
- **Humidity** - Create `HUMIDITY_VALUE` bindings for humidity sensor integration

Common Features:

- Global device availability monitoring via MQTT topic
- Device Available/Unavailable events for programming
- Configurable QoS and retain settings per item
- "Match one, default other" state parsing for flexible payload matching
- JSONPath extraction for nested JSON payloads
- All items persist across driver restarts

Installer Setup

Driver Installation

1. Ensure the MQTT Broker driver is installed and connected.
2. Add the "MQTT Universal" driver to your project.
3. In the Connections tab, bind the MQTT Universal driver to the MQTT Broker.

4. Configure global availability settings (optional).
5. Add devices and variables, then configure their MQTT topics.

Driver Setup

Driver Properties

Cloud Settings

Cloud Status (read-only)

Displays the DriverCentral cloud license status.

Automatic Updates [Off | *On*]

Turns on/off the DriverCentral cloud automatic updates. Changing it on any MQTT Broker or MQTT Universal item changes it on all of them.

Driver Settings

Driver Status (read-only)

Displays the state of the driver's link to the broker and of the device:

- `Connected` while the MQTT Broker is connected to the broker.
- `Disconnected` while it is not, or while the driver is not bound to an MQTT Broker.
- `Device unavailable` while the broker is connected and Availability Status reads `Unavailable`.

Driver Version (read-only)

Displays the current version of the driver.

Log Level [0 - Fatal | 1 - Error | 2 - Warning | *3 - Info* | 4 - Debug | 5 - Trace | 6 - Ultra]

Sets the logging level. Default is `3 - Info`. It returns to `3 - Info` when Log Mode turns off.

Log Mode [*Off* | Print | Log | Print and Log]

Sets the logging mode. Default is `Off`.

Support Code

Leave it blank unless Finite Labs support gives you a code: it opens Composer's Lua tab to this driver for an hour.

Cloud Monitoring [Disabled | *Enabled*]

When Enabled, the driver sends Finite Labs a short report once an hour, and a report of each error it hits, at most 10 a minute. We use them to find and fix problems in the driver. Set it to Disabled to send nothing.

A report holds the controller's MAC address and OS version, the driver's name, version, device ID and DriverCentral license status, how much memory the driver uses, and the sizes of its largest internal tables. An error report also holds the error message and where in the driver it happened. Before a message is sent, anything in it that looks like a password, token, key, network address or email address is removed.

The Diagnostics Snapshot action prints a report in the Lua output, and sends it only while Cloud Monitoring is Enabled.

Availability Settings

These settings track one availability topic for the whole driver.

Availability Topic

Sets the topic the driver subscribes to for the device's availability. Leave it empty if the device has no availability topic. Setting it adds the Device Available and Device Unavailable events, the availability conditional and the Availability Status variable. Set Payload Available, Payload Not Available or both to say which payloads mean what.

Changing or clearing it blanks Availability Topic Value, Availability Value Path Result and Availability Status until a payload arrives on the new topic. Clearing it keeps the events, the conditional and the programming on them.

Availability Topic Value (read-only)

Shown only when Availability Topic is set. Displays the last payload received on it.

Availability Value Path

Shown only when Availability Topic is set. Sets a JSONPath that picks the availability value out of a JSON payload. Leave it empty to use the whole payload. It takes the same paths as Value Path.

Availability Value Path Result (read-only)

Shown only when Availability Topic and Availability Value Path are set. Displays the value the path picked from the last payload, or `Error: Invalid JSON` or `Error: Path not found`.

Availability Status (read-only)

Shown only when Availability Topic is set. Displays `Available` or `Unavailable` once a payload matches.

Payload Available

Shown only when Availability Topic is set. Sets the payload that means the device is available. If only this is set, any other payload means unavailable. If both payloads are set, a payload that matches neither is ignored.

Payload Not Available

Shown only when Availability Topic is set. Sets the payload that means the device is unavailable. If only this is set, any other payload means available.

Add Devices

Each Add property takes the name of a new item. The item is not added, and the property shows an error for 2 seconds, when another item has the same name (`Error: Item with name '<name>' already exists`) or a variable the item would own is in use (`Error: Variable '<variable>' is already in use`). A variable is in use when another item owns it, when it is Availability Status, or when a property has its name. A run of spaces in a variable name counts as one underscore.

Add Relay

Enter a name to create a new relay. This creates a `RELAY` binding that can be connected to other Control4 devices or used in programming. The relay supports ON, OFF, TOGGLE, and TRIGGER commands.

Add Contact

Enter a name to create a new contact sensor. This creates a `CONTACT_SENSOR` binding that reports OPEN or CLOSED state based on MQTT messages.

Add Button

Enter a name to create a new button. Buttons publish a configured payload to an MQTT topic when triggered via the Press command.

Add Event

Enter a name to create a new event. Events subscribe to MQTT topics and trigger connected Control4 buttons/keypads when messages arrive.

Add Variables

These work as the Add Devices properties do.

Add String Variable

Enter a name to create a new STRING variable.

Add Bool Variable

Enter a name to create a new BOOL variable.

Add Number Variable

Enter a name to create a new NUMBER variable.

Add Float Variable

Enter a name to create a new FLOAT variable.

Add Temperature Variable

Enter a name to create a new TEMPERATURE variable. This creates a `TEMPERATURE_VALUE` binding that can be connected to thermostats.

Add Humidity Variable

Enter a name to create a new HUMIDITY variable. This creates a `HUMIDITY_VALUE` binding that can be connected to humidity sensors.

Manage Items

These properties are shown only once the driver has at least one item. Each entry in their lists reads `Name (TYPE) [ID]`.

Remove Item [*(Select)* | *added items...*]

> ⚠ Removing an item also removes its connection and the bindings made to it. Removing an event item also removes its event and the programming on it.

Selects an item to remove. The item's variables are removed with it.

Configure Item [*(Select)* | *added items...*]

Selects an item to show its settings under Item Configuration. A new item is selected here when you add it.

Item Configuration

These properties are shown when an item is selected in Configure Item. Which ones appear depends on the item's type and on the topics it has.

State Topic

Shown for every item except a button. Sets the topic the item subscribes to for its state or value.

State Topic Value (read-only)

Shown only when the item has a State Topic. Displays the last payload received on it.

Value Path

Shown only when the item has a State Topic. Sets a JSONPath that picks the value out of a JSON payload. Leave it empty to use the whole payload. The item's other settings then work on the value it picks.

A path starts with `$`, names nested fields with dots and picks an array entry by its index in brackets, counting from 0:

Payload	Value Path	Result
`{"temperature": 72.5}`	`$.temperature`	`72.5`
`{"sensors": {"temp": 68}}`	`$.sensors.temp`	`68`
`{"readings": [{"value": 42}]}`	`$.readings[0].value`	`42`
`{"data": [10, 20, 30]}`	`$.data[1]`	`20`

Value Path Result (read-only)

Shown only when the item has a State Topic and a Value Path. Displays the value the path picked from the last payload, or `Error: Invalid JSON` or `Error: Path not found`.

State (read-only)

Shown only for a relay or contact with a State Topic. Displays `On` or `Off` for a relay and `Open` or `Closed` for a contact. It stays empty until a payload matches.

State On

Shown only for a relay with a State Topic. Sets the payload that means the relay is on. If only this is set, any other payload means off. If both State On and State Off are set, a payload that matches neither is ignored. A new item starts with both empty, and the relay shows no state until one is set.

State Off

Shown only for a relay with a State Topic. Sets the payload that means the relay is off. If only this is set, any other payload means on.

Command Topic

Shown for a relay, a button, and a string, bool, number or float variable. Sets the topic the item publishes to.

Payload On

Shown only for a relay with a Command Topic. Sets the payload published when the relay turns on. A new item starts with it empty, so enter the payload your device expects: an empty field publishes an empty payload.

Payload Off

Shown only for a relay with a Command Topic. Sets the payload published when the relay turns off. Like Payload On, it starts empty.

Optimistic [*Auto* | Yes | No]

Shown only for a relay. Sets whether the relay's state changes as soon as it publishes a command. `Auto` does so only when the item has no State Topic, `Yes` always does, and `No` waits for the State Topic. Default is `Auto`.

State Open

Shown only for a contact with a State Topic. Sets the payload that means the contact is open. If only this is set, any other payload means closed. If both State Open and State Closed are set, a payload that matches neither is ignored. A new item starts with both empty, and the contact shows no state until one is set.

State Closed

Shown only for a contact with a State Topic. Sets the payload that means the contact is closed. If only this is set, any other payload means open.

Payload Press

Shown only for a button with a Command Topic. Sets the payload published when the button is pressed, by the Press command or by a device bound to its connection. A new item starts with it empty, and an empty field publishes an empty payload.

Event Type Filter

Shown only for an event. Lists the event types the event accepts, separated by commas. A message whose value, after any Value Path, matches none of them is ignored. Leave it empty to accept every message.

QoS [*0* | 1 | 2]

Shown only when the item has a State Topic or a Command Topic. Sets the MQTT quality of service level the item subscribes and publishes with. Default is `0`.

Retain [Yes | *No*]

Shown only when the item has a Command Topic. Sets whether the broker keeps the last message the item publishes for clients that subscribe later. Default is `No`.

Temperature Scale [*Celsius* | Fahrenheit]

Shown only for a temperature variable. Sets the scale of the values on the State Topic, so its connection reports the right temperature. Default is `Celsius`.

Connections

MQTT Broker (consumer)

Binds to the MQTT Broker driver, which carries every message the items send and receive. When you add the driver, it binds itself to the first MQTT Broker driver in the project.

{name} (provider, dynamic)

Adding a relay, contact, button, temperature variable or humidity variable adds a connection named for the item:

Item	Binding Class	Purpose
Relay	`RELAY`	Takes on and off commands and reports the relay's state
Contact	`CONTACT_SENSOR`	Reports whether the contact is open or closed
Button	`BUTTON_LINK`	Lets a bound device press the button
Temperature variable	`TEMPERATURE_VALUE`	Reports the temperature, in the item's Temperature Scale, when it changes
Humidity variable	`HUMIDITY_VALUE`	Reports the humidity when it changes

{name} (consumer, dynamic)

Adding an event adds a `BUTTON_LINK` connection named for the item. Each time the event fires, a device bound here gets a button press, as from a keypad button.

> **Note:** `{name}` is the item's name. A button's connection and an event's connection have the same binding class but point opposite ways: a button lets another device press it, and an event acts as a keypad button that drives the device bound to it.

Programming

Events

Event	Description
Device Available	Fires when the device changes from unavailable to available, including a change made while the controller was down. Added when Availability Topic is set, and kept when it is cleared.
Device Unavailable	Fires when the device changes from available to unavailable, including a change made while the controller was down. Added and kept with Device Available.
`{name} Triggered`	Fires when an event item receives a new message that passes its Event Type Filter.

Conditionals

Conditional	Description
Device availability status	Tests whether the device is `Available` or `Unavailable`. Added and kept with the availability events.

Commands

Command	Parameters	Description
Press	Button	Publishes the chosen button item's Payload Press to its Command Topic.

Variables

Variable	Type	Description
`{name}`	STRING, BOOL, NUMBER or FLOAT	A string, bool, number or float item's value from its State Topic. Writable: setting it publishes the new value to its Command Topic, and the next value on its State Topic replaces it.

Variable	Type	Description
`{name}`	FLOAT	A temperature or humidity item's last reading. Read-only.
`{name} State`	STRING	A relay's `On` or `Off`, a contact's `Open` or `Closed`, or an event's last event type. Read-only.
Availability Status	STRING	`Available` or `Unavailable`, or empty until a payload on the Availability Topic matches. Read-only.

> **Note:** `{name}` is the item's name. A temperature or humidity item's variable appears with its first reading. A State variable appears, empty, as soon as its item is added, and Availability Status as soon as an Availability Topic is set.

Developer Information



Copyright © 2026 Finite Labs LLC

All information contained herein is, and remains the property of Finite Labs LLC and its suppliers, if any. The intellectual and technical concepts contained herein are proprietary to Finite Labs LLC and its suppliers and may be covered by U.S. and Foreign Patents, patents in process, and are protected by trade secret or copyright law. Dissemination of this information or reproduction of this material is strictly forbidden unless prior written permission is obtained from Finite Labs LLC. For the latest information, please visit https://drivercentral.io/platforms/control4-drivers/utility/mqtt

Support

If you have any questions or issues integrating this driver with Control4 or MQTT, you can contact us at driver-support@finitelabs.com or call/text us at +1 (949) 371-5805.

Changelog

v20261106 - 2026-11-06

Added

- Added a `Reconnect` command to the MQTT Broker, so Control4 programming can reconnect it to the broker the same way as the `Reconnect` action.
- Added a `Support Code` property to the MQTT Broker and MQTT Universal drivers. A code from Finite Labs support opens the driver's Lua tab in Composer for an hour; leave it blank otherwise.
- Added a `Cloud Monitoring` property, Enabled by default, and a `Diagnostics Snapshot` action to every driver. Cloud Monitoring sends Finite Labs an hourly report of the driver's memory use and a report of each error it hits; the documentation says what they hold.

Fixed

- Fixed an MQTT Universal relay item running the bound relay's programming each time the controller restarted or the relay was bound. Programming now runs only when the relay opens or closes.
- Fixed an MQTT Universal contact item running the bound contact's programming each time the controller restarted, the driver was updated or the contact was bound. Programming now runs only when the contact opens or closes.
- Fixed MQTT Universal declaring its events again too early in its startup, which could leave an event item missing from Programming after a controller reboot until the driver was reloaded.
- Fixed an MQTT Universal event item firing again, running its programming and pressing a bound button, when the driver was updated, the MQTT Broker reconnected or an item's settings were changed. An event item now fires only when a new message arrives.
- Fixed MQTT Universal skipping the Device Available and Device Unavailable programming for an availability change made while the controller was down, and showing `Connected` for a device that was already unavailable when the driver started.
- Fixed clearing MQTT Universal's `Availability Topic` deleting the Device Available and Device Unavailable events and the programming on them. The events now stay, and the availability readings go blank.

- Fixed MQTT Universal's relay, contact and event State variables and its Availability Status variable missing from programming until the first message arrived. They now appear, empty, as soon as the item is added or the `Availability Topic` is set.
- Fixed removing an MQTT Universal temperature or humidity item leaving its variable behind.
- Fixed MQTT Universal adding an item whose variable was already in use by another item, by Availability Status or by a property, so the two overwrote each other. The item is now refused, and the Add property says which variable is in use.
- Fixed MQTT Universal temperature and humidity items sending bound devices the same reading again with every message. A reading now goes out only when it changes.
- Fixed an MQTT Universal variable item keeping a value from Control4 programming that the device refused. The variable now shows the device's value again at its next report.
- Fixed MQTT Universal's `6 - Ultra` logging not starting when `Log Mode` was turned on and not stopping when it turned off. `Log Level` now also returns to `3 - Info` when logging turns off, as on the MQTT Broker.
- Fixed MQTT Universal sometimes not finishing its start when one of its settings hit an error while loading.
- Fixed the MQTT Broker printing its password to the Lua output when debug printing was on.
- Fixed the MQTT Broker waiting 30 seconds to reconnect after every drop. It now tries again 2 seconds after a drop and waits longer after each failed attempt, up to 30 seconds.
- Fixed the MQTT Broker staying at `Connecting...` with no retry when its `Broker Address` did not resolve, and showing a bare `Disconnected` while it retried. `Driver Status` now shows why the connection dropped or failed, and the driver retries.
- Fixed the MQTT Broker's Broker Disconnected event firing again at each failed retry while the broker stayed down, and not firing when a Reconnect or a settings change closed a live connection. Broker Connected and Broker Disconnected now fire only when the connection changes.
- Fixed updating or removing the MQTT Broker leaving its old connection to the broker open.
- Fixed a licensed MQTT Broker staying disconnected until Reconnect when it lost the broker while the DriverCentral Cloud driver was not answering. It now reconnects on its usual retry.

Changed

- Changed MQTT Universal's Composer category from Misc to Utility, so it is listed beside the MQTT Broker.
- Changed Automatic Updates to one setting for the MQTT Broker and MQTT Universal drivers: changing it on any of them now changes it on all of them.
- Without an active license or trial, the MQTT Broker now disconnects five minutes after it loads and shows `No active license`. It connects again as soon as a license is active, with no Reconnect needed.

v20261024 - 2026-10-24

Fixed

- Fixed an MQTT event's button link acting on one press twice, which turned a bound on/off light straight back off and left a dimmer off instead of ramping it. Event button links now send the same command pair a Control4 keypad sends for a tap.
- Fixed the Device Available, Device Unavailable and event item Triggered events leaving the device's name out of their script title in Composer Pro. The title now names the device.
- Fixed MQTT Universal temperature sensor items not updating a bound thermostat.
- Fixed MQTT Universal sensor and number variable items accepting a `nan` or `inf` payload, such as a failed probe publishes. The last good value is kept.

v20260415 - 2026-04-15

Fixed

- Fixed MQTT Broker auto-reconnect reliability.

v20260414 - 2026-04-14

Fixed

- Fixed MQTT Universal variables losing their programming "Set Variable" callback after the first inbound MQTT message on the variable's state topic.
- Fixed MQTT Universal BOOL variables not toggling reliably from programming due to incorrect serialization to Control4.

- Fixed MQTT Universal variables losing their writable state after a controller reboot or driver reload.

v20260217 - 2026-02-17

Changed

- Temperature conversions (Fahrenheit/Celsius) now use single decimal precision instead of rounding to the nearest integer or half degree.

Fixed

- Fixed variable restore order during driver initialization on controller reboot.

v20260127 - 2026-01-27

Fixed

- Fixed incorrect DriverCentral product ID in MQTT Universal driver.

v20260120 - 2026-01-20

Fixed

- Fixed consumer bindings (e.g., BUTTON_LINK for Event entities) not restoring their connections on controller reboot. The bindings module now tracks connection info and automatically re-establishes consumer binding connections when bindings are restored.

v20260119 - 2026-01-19

Added

- Added Event entity type to MQTT Universal driver for receiving MQTT events and triggering Control4 buttons/keypads.

Changed

- Updated MQTT Broker documentation to include Programming section with events and conditionals.

- Updated MQTT Universal documentation to include missing read-only properties and BUTTON_LINK binding information.

Removed

- Removed deprecated MQTT Switch, MQTT Contact, and MQTT Button drivers.

v20260117 - 2026-01-17

Added

- Added the MQTT Universal driver for managing multiple MQTT devices from a single driver instance. Supports relays, contacts, buttons, variables, and sensors with features including:
  - Keypad button linking for MQTT buttons
  - Device availability conditionals and events
  - State variables for relay and contact items
  - JSONPath value extraction for complex JSON payloads
- Added "Broker Connected" conditional and events to the MQTT Broker driver.

Deprecated

- Deprecated the MQTT Switch, MQTT Contact, and MQTT Button drivers. Use MQTT Universal instead. This is the terminal release for these drivers.

v20251229 - 2025-12-29

Added

- Initial Release.