

AD-ÇA.ca, MQTT

Plugin Description

The [AD-DA.ca](#) MQTT plugin provides MQTT 3.1.1 send and receive workflows for Q-SYS systems. It supports broker connection, topic publishing, topic subscriptions, message history, and optional JSON message outputs.

This plugin is intended for operational MQTT control and monitoring tasks where reliable topic exchange, simple setup, and clear runtime feedback are required.

This plugin currently supports MQTT over TCP only. MQTT over WebSocket (WS/WSS) is not supported at this time.

Configuration Overview

Plugin version

This document is written for the plugin:

- [AD-DA.ca](#) MQTT

QDS compatibility

This plugin has been developed and tested on:

- Q-SYS Designer version 10.4

Release note

2026-07 : 1.0.0.0

- Initial Release

Properties

License

Type: String **Default value:** *Empty* **Description:** License key activated for the main core.

Licenses are issued on a "per core" basis, as in each core needs its own license to run as many instances of the plugin as needed on this particular core and licenses are not transferable from core to core.

License keys can be added while in Offline mode, in the Properties pane on the right. To purchase licenses, go to <https://ad-da.ca>.

The plugin can be emulated for free, allowing you to prepare everything without requiring you to have access to a core.

License Backup

Type: `String` **Default value:** `Empty` **Description:** License key activated for the backup core.

If your main Q-SYS Core fails, the backup Core automatically takes over without needing a license to function as a temporary replacement. However, if the design boots from the backup Core, a valid license is required. Additionally, if the backup Core is restarted (due to power loss or other reasons), the plugin instances cannot initialize without a valid license.

Mode

Type: `Enum` **Choices:** `Send`, `Receive` **Default value:** `Send` **Description:** Selects plugin behavior between publish workflows and subscribe workflows.

Topic Count

Type: `Integer` **Range:** `1 to 32` **Default value:** `4` **Description:** Number of topic rows exposed on the Controls page.

Variable Count

Type: `Integer` **Range:** `1 to 16` **Default value:** `4` **Description:** Number of template variables exposed in Send mode.

JSON Output

Type: `Boolean` **Default value:** `false` **Description:** Enables JSON output per Receive row, including `topic`, `payload`, and UTC `timestamp`.

Dev

Type: `String` **Default value:** `Empty` **Description:** Development option.

Debug Print

Type: `Enum` **Choices:** `None`, `Tx/Rx`, `Tx`, `Rx`, `Functions Call`, `All`, `Verbose` **Default value:** `None` **Description:** Controls runtime debug logging verbosity.

Controls

Global Control

STATUS

Control name: Status

Type: Indicator

Controls Pin: Output

Description: Displays plugin health and connection state.

Typical values:

OK: Connected or disconnected without error.

FAULT: Connection/configuration/license issue.

Controls Page (Send Mode)

SEND SETTINGS SECTION

AUTO SEND

Control name: Auto Send

Type: Button Toggle

Controls Pin: Input

Description: In Send mode, publishes automatically when Topic N, Payload N, or Var N changes.

TOPICS SECTION

TOPIC N

Control name: Topic N

Type: Text Input

Controls Pin: Both

Description: Topic template for row N.

PAYLOAD N

Control name: Payload N

Type: Text Input

Controls Pin: Both

Description: Payload template for row N.

SEND N

Control name: Send N

Type: Button Trigger

Controls Pin: Input

Description: Sends the current Topic N and Payload N immediately.

Controls Page (Receive Mode)

TOPIC N

Control name: Topic N

Type: Text Input

Controls Pin: Both

Description: Subscription topic filter for row N.

RECEIVED N

Control name: Received N

Type: Indicator LED

Controls Pin: Output

Description: Briefly flashes when an incoming message matches row N.

INCOMING TOPIC N

Control name: Incoming Topic N

Type: Text Output

Controls Pin: Output

Description: Exact topic string of the last message matched by row N.

PAYLOAD N

Control name: Payload N

Type: Text Output

Controls Pin: Output

Description: Payload of the last message matched by row N.

JSON N

Control name: JSON N

Type: Text Output

Controls Pin: Output

Description: Optional JSON output for row N containing topic, payload, and UTC timestamp.

Variables Page (Send Mode Only)

VAR N

Control name: Var N

Type: Text Input

Controls Pin: Both

Description: Variable value used to resolve Send mode placeholders like {1}.

History Page (Receive Mode Only)

CLEAR HISTORY

Control name: Clear History

Type: Button Trigger

Controls Pin: Input

Description: Clears all entries shown in History List.

HISTORY LIST

Control name: History List

Type: Text Output

Controls Pin: Output

Description: Receive history text output, with newest message at the top.

Maximum history size is 100 entries.

Oldest entries are removed when the limit is exceeded.

Setup Page

ETH

Control name: Eth

Type: Text ComboBox

Controls Pin: Input / Output

Description: Selects which local interface/IP the plugin uses for outbound TCP connection.

IPADDRESS

Control name: IpAddress

Type: Text Input

Controls Pin: Input / Output

Description: Broker host or IP address to connect to.

PORT

Control name: Port

Type: Text Input

Controls Pin: Input / Output

Default value: 1883

Description: Broker TCP port for MQTT.

CLIENTID

Control name: ClientId

Type: Text Input

Controls Pin: Input / Output

Description: MQTT client identifier sent during CONNECT.

USERNAME

Control name: Username

Type: Text Input

Controls Pin: Input / Output

Description: Optional MQTT username for authenticated brokers.

PASSWORD

Control name: Password

Type: Text Input

Controls Pin: Input / Output

Description: Optional MQTT password for authenticated brokers.

CONNECT

Control name: Connect

Type: Button Toggle

Controls Pin: Input / Output

Description: Starts connection when ON and disconnects when OFF.

CONNECTED

Control name: Connected

Type: Indicator LED

Controls Pin: Output

Description: On while broker connection is active.

Appendix

Send Mode Help

- Topic N and Payload N are templates sent by Send N.
- Auto Send publishes automatically when topic, payload, or variable values change.
- Template variables are defined in the Variables page (Var 1..N).
- Supported placeholders: {1}, {2}, {3} and {var1}, {var2}, {var3}.

Examples:

- Topic: building/{1}/set
- Payload: level/{1}
- Payload JSON: {"value": "{2}", "source": "qsys"}
- Topic with multiple vars: cmd/{1}/{2}/{3}

Receive Mode Help

- Topic N is a subscription filter.
- Incoming Topic N shows the exact matched topic.

- `Payload N` shows the matched message payload.

Wildcard rules:

- `+` matches exactly one topic level.
- `#` matches all remaining levels and must be the last level.
- No wildcard means exact topic match only.

Examples:

- `#` listens to everything.
- `site/#` listens to all topics under `site/`.
- `site+/temperature` matches one level between.
- `site/room1/temp` matches only that exact topic.
- `site+/temp/#` is valid mixed wildcard usage.
- `site/#/temp` is invalid because `#` must be last.

Transport Note

- Current transport support is MQTT over TCP only.
- MQTT over WebSocket (WS/WSS) is currently not supported.