

Inhaltsverzeichnis

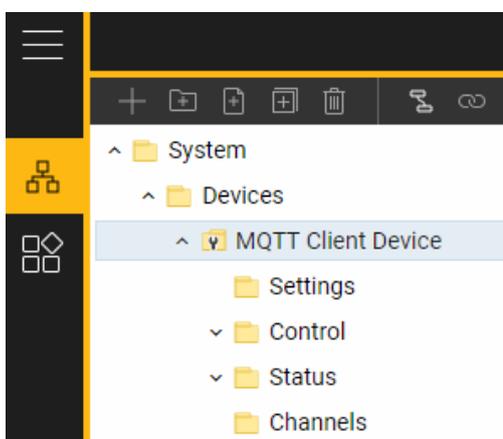
- Konfiguration** 1
 - Channel 1
 - Settings 1
 - Hinzufügen eines Channels 2
 - Topics 3
 - Path 4
 - Abonnieren 4
 - Veröffentlichen 4
- Fehlerdiagnose** 4
 - Kanal 4
 - Topic 5
 - Logdatei 5
- Entities** 5
 - Device 5
 - Channel 6
- Ordner & Dateien** 6
 - Ordner 6
 - Dateien 6
- Versionsinformation** 6
 - Dieses Dokument 6
 - Plugin 7
 - Assembly 7

MQTT Client Device Plugin

Das MQTT Client Device Plugin ermöglicht das Veröffentlichen und Abonnieren (engl. Publish/Subscribe, kurz Pub/Sub) von Werten anhand von Topics in Geräten, die über einen **MQTT Broker** verfügen.

Konfiguration

Die gesamte MQTT Client Device Plugin-Konfiguration befindet sich unter dem Nodepfad /System/Devices/MQTT Client Device.



Channel

Ein MQTT Client Device Channel repräsentiert die Verbindung zu einem MQTT Broker.

Settings

Client Id

Ein frei wählbarer Bezeichner der es erlaubt im Broker den Client zu identifizieren.

Server Address

Die URL zum MQTT Broker, stets mit Schema 'mqtt.tcp' gefolgt vom Hostname oder IP Adresse des Hosts und schließlich der zu verwendende Port z.B.
mqtt.tcp://mymqttbroker:1883.

Login Type

Definiert, welche Authentifizierungsart verwendet wird.

Login Name

Der Benutzer des MQTT Brokers, der für die Authentifizierung benutzt wird.

Login Password

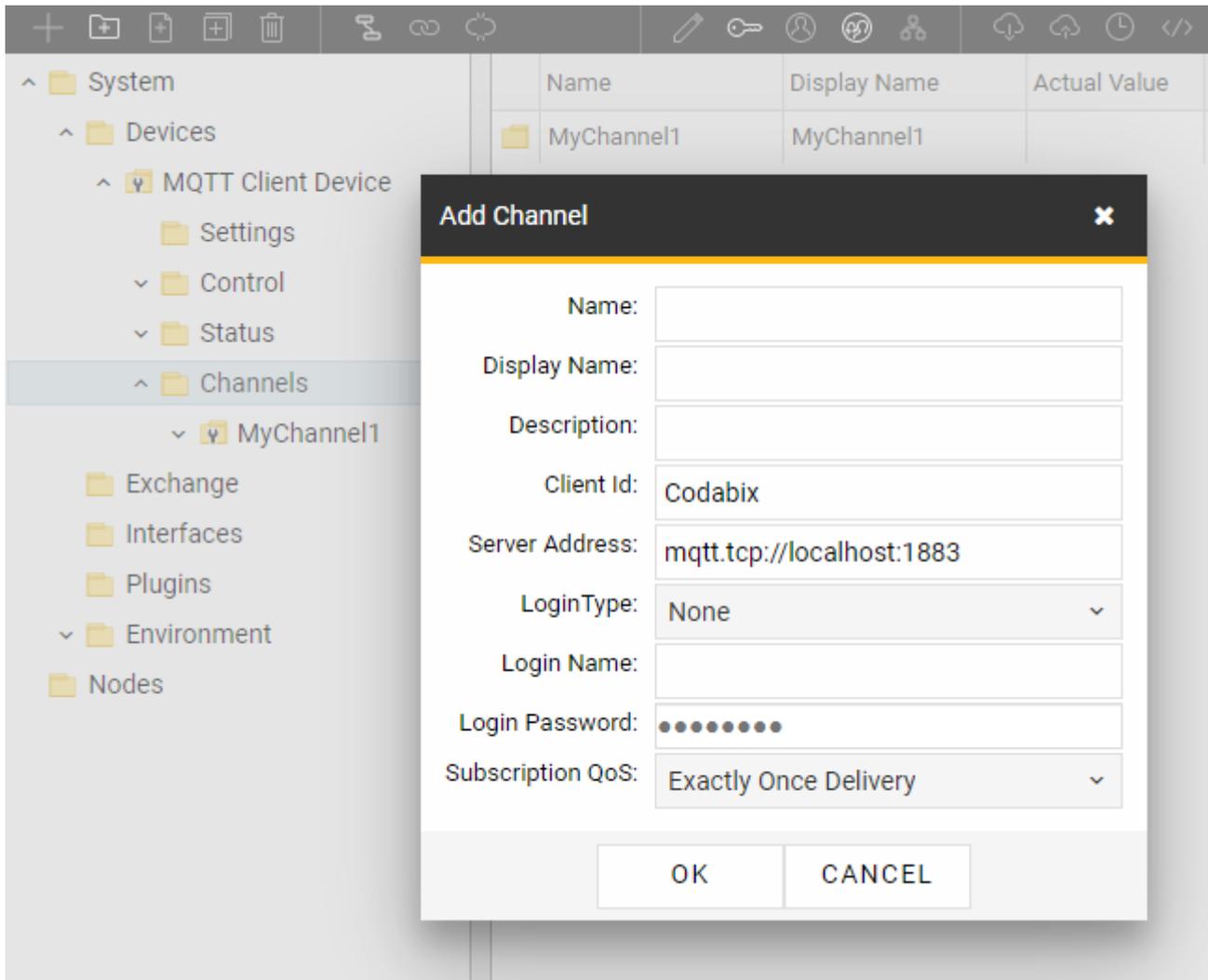
Das Passwort des Benutzers des MQTT Brokers.

Subscription QoS

Die Qualität des Services (engl. Quality of Service, kurz QoS) der verwendeten Subscription zur Aktualisierung der Wert der Topics. Der Wert bestimmt die Zustellgarantie für die Aktualisierungs-Nachricht vom Broker zum Client. Die folgenden Werte sind möglich:

- **At Most Once Delivery:** Die Nachricht wird nie, einmal oder mehrmals zugestellt.
- **At Least Once Delivery:** Die Nachricht wird mindestens einmal (oder öfter) zugestellt.

Hinzufügen eines Channels



Um einen neuen MQTT-Channel zu erstellen, gehen Sie wie folgt vor:

1. Fügen Sie einen Folder Node unter dem Node MQTT Client Device/Channels hinzu, oder machen Sie einen Rechtsklick auf den MQTT Client Device/Channels-Node und wählen Sie Add Channel aus.
2. Tragen Sie im Add Channel-Dialog die Settings für die MQTT-Verbindung ein.
3. Nachdem Sie „Save“ geklickt haben, wird die Channel-Node erstellt.
4. Sie können den Kanal starten, indem Sie die Channel-Node auswählen und den Startbutton klicken.

Topics

Unter dem Topics-Node können Sie Datenpunktnodes erstellen, die auf dem MQTT Broker abonniert beziehungsweise an die Werte veröffentlicht werden sollen.

Unterstützte Node-Werttypen:

- String (je nach Broker als JSON)

Sie können auch Folder-Nodes erstellen, um Datapoint-Nodes zu gruppieren.

Path

Die Path-Property des Nodes wird verwendet, um den Namen des Topics im Broker festzulegen.

Abonnieren

Ein einmal erstellter Topic-Node ändert seinen Wert durch ein aktives Abonnement (engl. Subscription). Hierdurch übermittelt der MQTT-Broker jede Änderung des Topics an den Node, der wiederum diesen Wert als aktuellen Node-Wert übernimmt.

Veröffentlichen

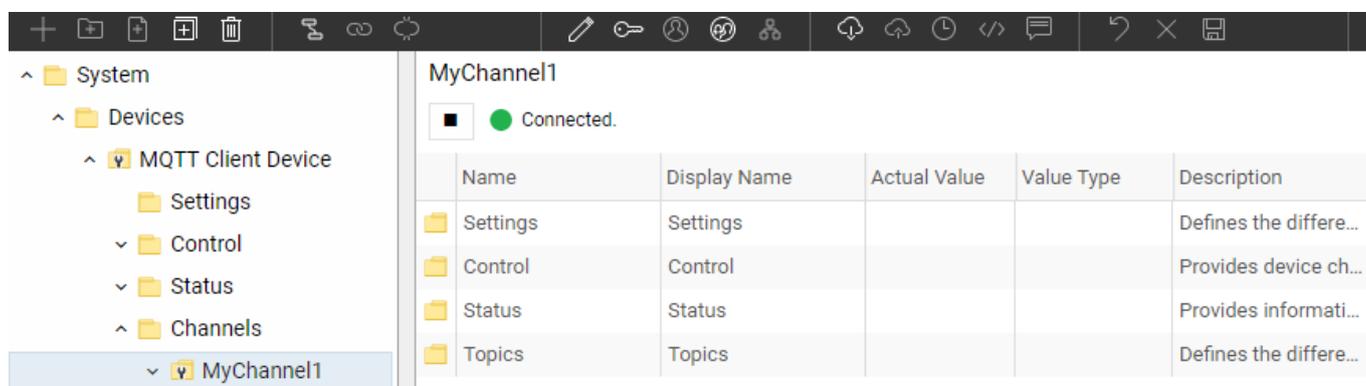
Wenn Topic-Nodes geschrieben werden, führt das Plugin eine Veröffentlichung (engl. Publish) des Wertes durch.

Fehlerdiagnose

Das MQTT Client Device Plugin liefert je nach zu untersuchender Schicht verschiedene Statusinformationen. Generell werden die kanalbasierten Diagnoseinformationen beim Verbindungsauf-/abbau, produziert. Die Topic-basierten Diagnoseinformationen werden während eines aktiven Abonnements beziehungsweise nach einen Schreibzugriff auf den verschiedenen Topics produziert.

Kanal

Um den Status des MQTT-Kanals zu überwachen und zu diagnostizieren, werfen Sie einen Blick auf das folgende Bild:



Das obige Bild zeigt das Bedienfeld des MQTT-Kanals, das alle statusrelevanten Informationen anzeigt. Das Bedienfeld aktualisiert automatisch seine Statusinformation, wenn ein neuer Status verfügbar ist.

Statuskreis

Farbe	Bedeutung
	Der Kanal ist gestoppt. Klicken Sie den ►-Button, um ihn zu starten.
	Der Kanal startet oder stoppt gerade oder wartet auf den Verbindungsaufbau.
	Der Kanal ist bereit für Lese-/Schreiboperationen. Sie ihn können durch Klick auf den ■-Button stoppen.
	Der Kanal läuft, aber die Verbindung ist momentan fehlerhaft. Bitte überprüfen Sie den Statustext für weitere Informationen.

Topic

Um den Status der verschiedenen Topics zu überwachen und zu diagnostizieren, werfen Sie einen Blick auf die in Codabix angezeigte Status-Eigenschaft der Spalte. Aktivieren Sie historische Werte oder stellen Sie ein aktives Abonnement auf den Topics sicher, um die vom Broker bereitgestellten Werte und das Ergebnis in den Topics zu speichern.

Name	Display Name	Actual Value	Value Type	Description	Path	Status
 A	A	abc	String		a/1	Bad: Operation timed-out.
 B	B	456	String		b/2	Bad: Operation timed-out.

Logdatei

Alle kanalbezogenen Statusinformationen werden auch in die kanalspezifische Logdatei im [LoggingFolder] protokolliert. Jede Logdatei wird nach dem Namensschema MQTT Client Device.<ChannelName>.log benannt.

Der Inhalt einer solchen Logdatei kann wie folgt aussehen:

```
...
2018-04-11 11:32:37.0 +2: [Error] Error (Severity=High): Code=[-1],
Text=[The operation has timed-out.], Details=[]
...
```

Entities

Wie jedes Device Plugin erweitert das MQTT Client Device Plugin das Codabix [Device Modell](#).

Device

Der Device Typ `MqttClientDevice` des Plugins definiert auch den `MqttClientDeviceChannel` und erweitert somit die grundlegenden `CodabixDevice` und `CodabixDeviceChannel` Entities. Während das `MqttClientDevice` nur eine Konkretisierung des `CodabixDevice` darstellt, erweitert der `MqttClientDeviceChannel` den `CodabixDeviceChannel` mit den MQTT Topic Entities.

Channel

Jeder Kanal definiert seinen eigenen ein- und ausgehenden Datenfluss, um die Werte der Topics entsprechend dem Pub/Sub-Modell mit dem MQTT Broker auszutauschen.

Der für den lesenden Zugriff verantwortliche Datenfluss liefert standardmäßig keine Werte. Auch nicht, wenn ein Anwender oder Plugin in Codabix einen synchronen Lesevorgang der Channel-Topics anfordert (z.B. mit der „Read actual value“-Funktion in der Codabix Webkonfiguration). Da, im Sinne des Pub/Sub-Modells, neue Werte vom MQTT Broker an den MQTT Client gesendet werden, ist ein explizites Lesen nicht möglich. Wird ein Wert durch den eingehenden Datenfluss empfangen (mittels Subscription), wird dieser in die entsprechenden Codabix-Nodes geschrieben.

Der ausgehende Datenfluss wiederum schreibt die Werte in den Topic des Brokers (mittels Publish), wenn ein Client oder Plugin Werte in die Channel-Topics schreibt.

Damit der Wert eines MQTT Topics auf dem aktuellen Stand bleibt, können Sie in der Webkonfiguration bei dem Node „History Options“ auf Yes stellen (was eine interne Subscription erstellt), oder Sie können zum Beispiel einen OPC UA Client verwenden, der mit dem OPC UA Server Plugin verbunden ist und damit eine Subscription für die MQTT-Topicnodes erstellen. In diesen Fällen verwendet der Channel den eingehenden Datenfluss dazu, um über eine Subscription zum Broker, neue Werte zu empfangen und schreibt diese bei Wertänderung automatisch in den entsprechenden Codabix-Node.

Ordner & Dateien

Ordner

Name	Pfad	Zweck / Verwendung
AssemblyFolder	<CodabixInstallDir>/plugins/MqttClientDevicePlugin/	Beinhaltet die Plugin-Assemblydatei.
ConfigFolder	<CodabixProjectDir>/plugins/MqttClientDevicePlugin/	Beinhaltet die Plugin-Konfigurationsdatei.
LoggingFolder	<CodabixProjectDir>/log/	Beinhaltet die Plugin-Logdateien.

Dateien

Typ	Pfad	Zweck / Verwendung
Assembly	[AssemblyFolder]/Codabix.MqttClientDevicePlugin.dll	Die Plugin-Assembly-Datei.
Logging	[LoggingFolder]/MQTT Client Device.<ChannelName>.log	Die Logdatei.

Versionsinformation

Dieses Dokument

Datum	2021-02-05
--------------	------------

Version	1.0
----------------	-----

Plugin

Name	MQTT Client Device Plugin
Node	/System/Devices/MQTT Client Device
Version	1.0.0

Assembly

Name	Codabix.MqttClientDevicePlugin.dll
Datum	2021-02-05
Version	1.10.0.0

From:
<https://www.wiki.codabix.com/> - **CoDaBix®**

Permanent link:
<https://www.wiki.codabix.com/de/plugins/device/mqttclientdeviceplugin>

Last update: **2023/11/17 10:33**