

MQTT SparkplugB

For a generic explanation of the configurations for the Device Module, Channels, Nodes, and Points, please refer to the reference guide.

Summary Information

Communication Driver Name: MQTTspB **Implementation DLL:** T.ProtocolDriver.MQTTspB.dll

Interface: TCPIP

Protocol: MQTT (Message Queuing Telemetry Transport) is a machine-to-machine (M2M)/"Internet of Things" connectivity protocol. "Sparkplug provides an open and freely available specification for how Edge of Network (EoN) gateways or native MQTT enabled end devices and MQTT Applications communicate bi-directionally within an MQTT Infrastructure." [SparkplugB namespace and Payload format](#).

PC Hardware Requirements: Ethernet Board.

Implemented Methods: Connect, Disconnect, Subscribe, and Publish.

Channel Configuration

Protocol Options

- **Type:** Defines how the channel works when connecting to the MQTT Broker.
 - **Application Node:** It is a MQTT client that consumes real-time messages, performs analytical calculations, saves history data, etc. Subscription is allowed but publishing is not allowed.
 - **Scada IIoT Host:** It is a primary MQTT client that consumes real-time messages but also publishes commands (messages) to [EoN Node](#) nodes and devices.
 - **EoN Node:** It is a MQTT client that mainly functions by publishing messages read from field. It also can receive commands from a [Scada IIoT Host](#).
- **ClientID:** Indicates the client identification used for connecting to the MQTT Broker.
- **Tag Properties:** Defines the tag properties beyond the [Value](#). The names of the properties should be separated by a ',' (comma).
- **Time Publish Rate:** Indicates the time for publishing in ms. The messages that will be published to the MQTT Broker are grouped and sent using this period.



The Built-in Broker can be started using the [Run Broker](#) button.

Node Configuration

Station Configuration

The station syntax is:

BrokerURL ; Port ; [Username] ; [Password] ; [X509Certificate] ;

[SslProtocol] ; [Websocket] ; [ScadaIIoTHostID]

Where:

- **BrokerURL:** MQTT Broker (Server) URL.
- **Port :** MQTT Broker port. It must be the same port as configured in the Broker. Default value is 1883.
- **[Username] :** Username defined on the MQTT. This is requested if the Broker needs this configuration (optional).
- **[Password] :** Password defined on the MQTT. This is requested if the Broker needs this configuration (optional).
- **[X509Certificate] :** Path of the X509 Client. If using a server-side only certificate, this field does not need to be configured. This must be the complete path of the X509 certificate on the client computer. The certificate must be installed on the computer. Note: One of the easiest ways to install the certificate on the client computer is to use the wizard and use the "Internet Options" to import the certificate in the "Trusted Root Certification Authorities". You need to import the certificate in the DER format (optional).
- **[SslProtocol] :** If MQTT Broker works by using a certificate, this field is mandatory and must be the same as configured in the MQTT Options: None, Ssl2, Ssl3, Tls, Tls11, and Tls12 (optional).

- **[WebSocket]:** Flag indicating if the connection to the MQTT Broker should be via the WebSocket. Check how MQTT Broker works before checking it (optional).
- **[ScadaIIoTHostID]:** If the channel works as EoN Node, some Scada IIoT Host are active, and other MQTT Brokers are also present in the network (redundancy, scalability, etc), then this field is used to know what MQTT Broker must be used. Scada IIoT Host publishes your ID when connecting to the MQTT Broker (optional).



All nodes of the same channel share all the device points regardless of which node they were configured for. Setting more than one node for the same channel should be used for MQTT Brokers redundancy.

Point Configuration

Address

The syntax for the MQTT communication points are:

?GroupId?;?EdgeNodeId?;?DeviceId?;?QoS?

Where:

- **GroupId:** Provides a logical grouping of Edge Nodes. Wildcards are not supported.
- **EdgeNodeId:** Identifies ID of Edge Nodes. Wildcards are not supported.
- **QoS :** Quality of Service (QoS) is an agreement between sender and receiver of a message regarding the guarantees of delivering a message. There are 3 QoS levels:
 - AtMostOnce
 - AtLeastOnce
 - ExactlyOnce



For the EoN Node type of Channel, the edge node ID element of the Sparkplug Topic Namespace uniquely identifies the MQTT EoN node within the infrastructure.

The group ID/Edge Node ID element must be unique from any other group ID/Edge Node ID assigned in the MQTT infrastructure.

This means that each Client (Instance of **Devices > Channel**) can have only one Edge Node ID, but more than one device ID assigned.

TagProvider

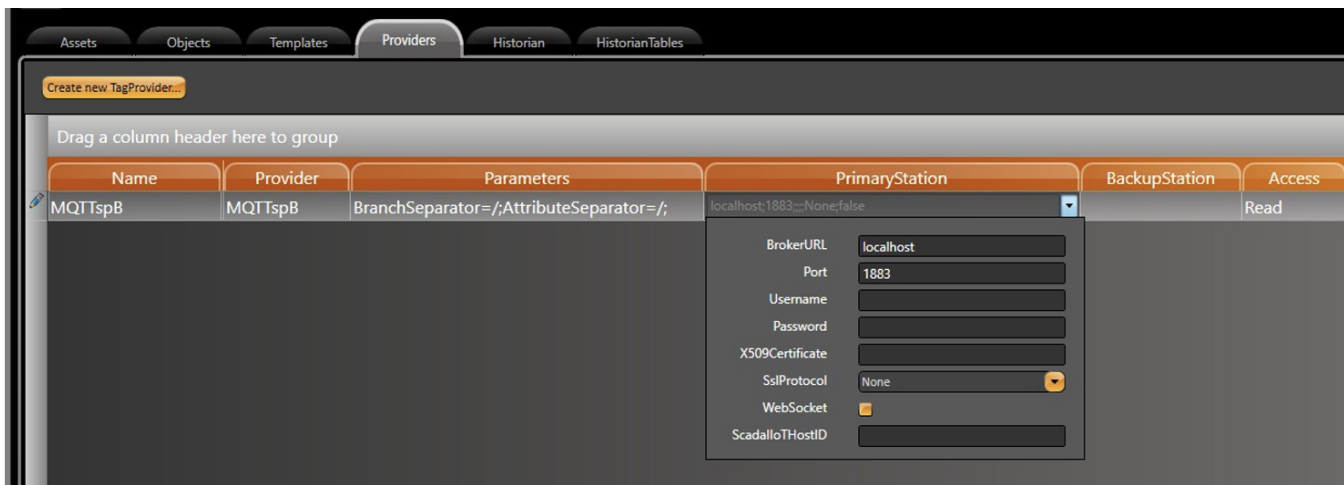
This Communication Protocol supports the TagProvider feature, which is a tool that allows you to access your Communication Device Data Model without creating any Project Tags.

For more information, please refer to the [TagProvider](#) document.

How to Configure

To configure the MQTTspB protocol as a TagProvider, navigate to **Edit > Tags > Providers**, and create a new provider for the MQTT +SparkplugB protocol.

Configure the items under the **PrimaryStation** column the same way that was described in the Node Configuration.



The parameters configured in the **Channel** tab are not configured in this TagProvider feature. The default configuration is described below:

- **Type:** Application Node.
- **ClientID:** MQTTspB Util prefix followed by a new GUID structure name.
- **Tag Properties:** The value property is used in this communication.
- **Time Publish Rate:** 100ms.



The TagProvider for the MQTTspB protocol is Read-Only, meaning you cannot publish data.

Simulator

MQTTspB Simulator

Connection String Parameters

Broker URL:

127.0.0.1

Port:

1883

Username:

Password:

Disconnect

Connect

Simulaton Settings

Base GroupID Name:

GroupID

Base NodeID Name:

NodeID

Number Of Nodes:

1

Base DeviceID Name:

DeviceID

Number Of Devices per Node:

1

Apply

Runtime Simulation Settings

Send one Data Sample

Send Continuous Data

Show Log

Status:

Troubleshoot

The status of the driver execution can be observed through the diagnostic tools, which are:

- Trace window
- Property Watch
- Module Information