MQTT SparkPlug

This document contains information on how to access elements from an MQTT Broker data model in any area of the Engineering Environment, and it contains information on how to access the entire, or just a part, of the Asset Model Structure.

System Requirements

- FactoryStudio FrameworX: Version fs-9.1.12
- Devices > Channel and Node created for the MQTTspB protocol

Device Configuration

On Device > Channel, create a new Channel for the MQTTspB protocol. Configure the ProtocolOptions field with the necessary information. Make sure the Type is set for Application Node.

To connect to your Broker, go to Device > Node, create a node, assign it to the newly created channel, and fill the Station parameters.

There is no need to create communication Points.

Unified Namespace

Context

Unified Namespace is a software solution that acts as a centralized repository of data, information, and context where any application or device can consume or publish data needed for a specific action.

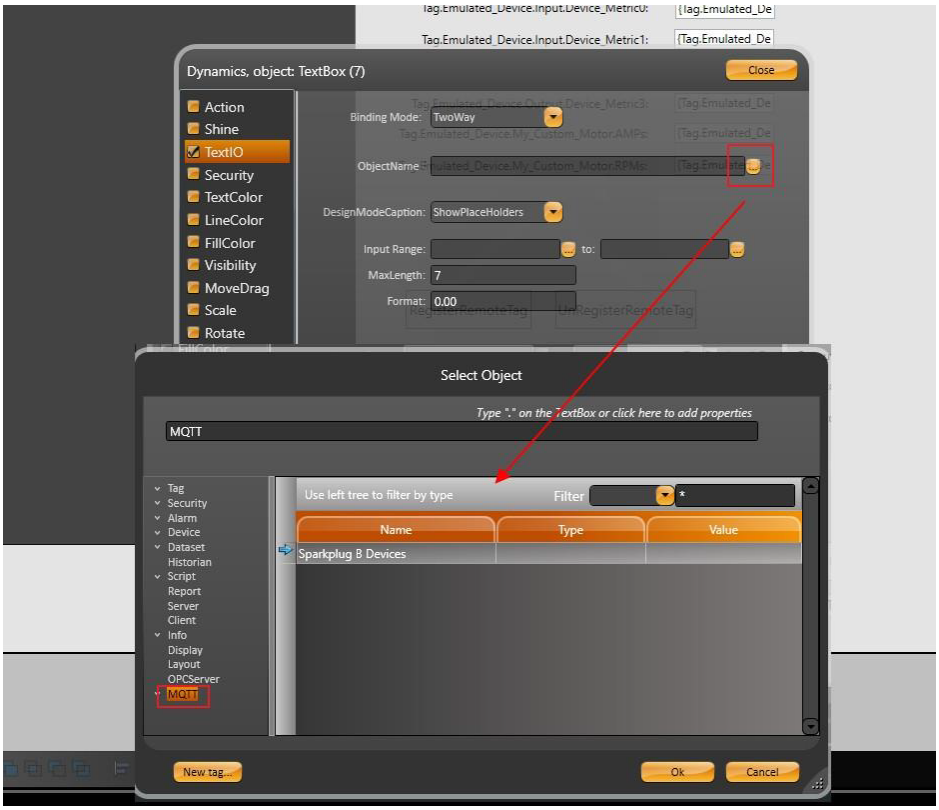
It allows users to collect data from various sources and transform it to a format that other systems can understand. Without a centralized data repository, it could take months to deploy a new analytics application across the entire enterprise versus hours with a unified namespace.

Access Elements in Engineering

Once you have a successful connection established to an MQTT Broker, you can start using your Data Model's available variables in your Project.

In the Draw Environment, insert a TextBox element in your display. Double-click on it and add a TextIO Dynamic.

In the ObjectName field, select the Browse button. In the dialog window, you should see a list of objects you can choose from. MQTT will be at the bottom of the list.

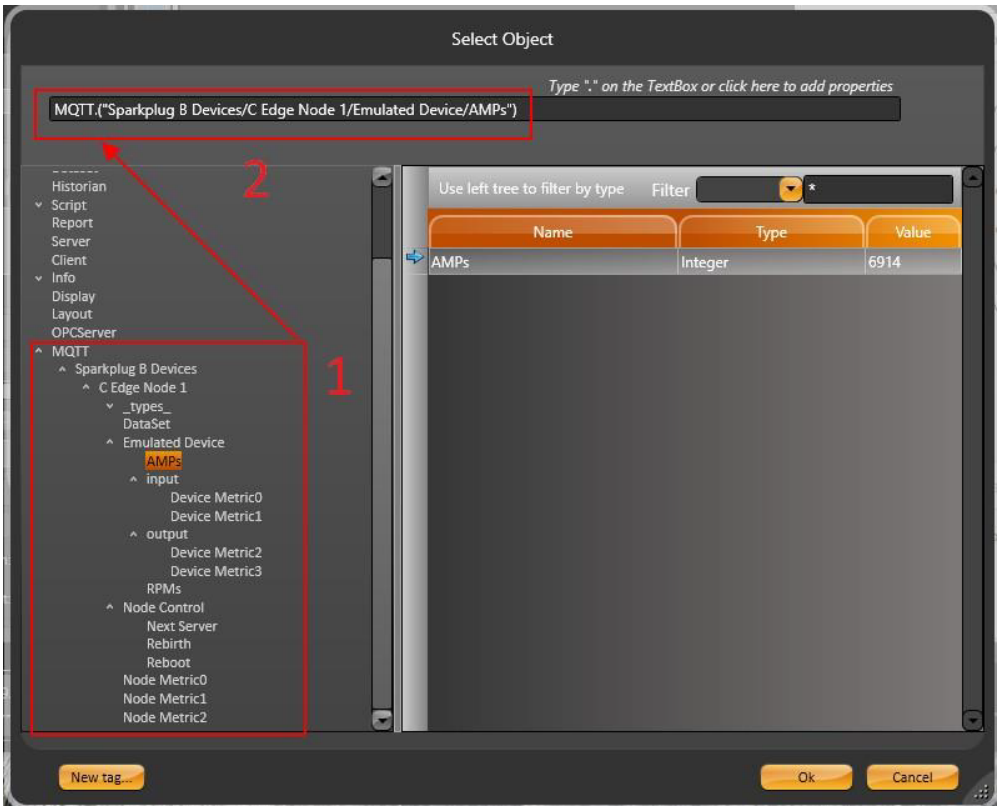


You can click on MQTT to expand it and browse through the existing elements that is connected to your Broker (1).

The expression field will be filled with a syntax.

Eample: MQTT.("? Address In MQTT ?") (2).

By doing this, you can add the information from this Communication Protocol directly to your display, without needing to create a Tag and Communication Point.



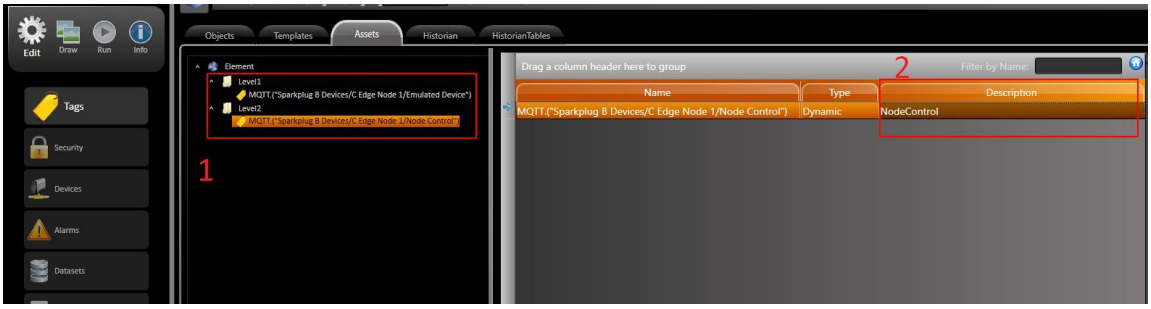
Also, this feature can be used to create AlarmItems or to store data in Historian Tables. You can use this MQTT variable as a Communication Point to Write Data in a different Protocol.

Asset Modeling

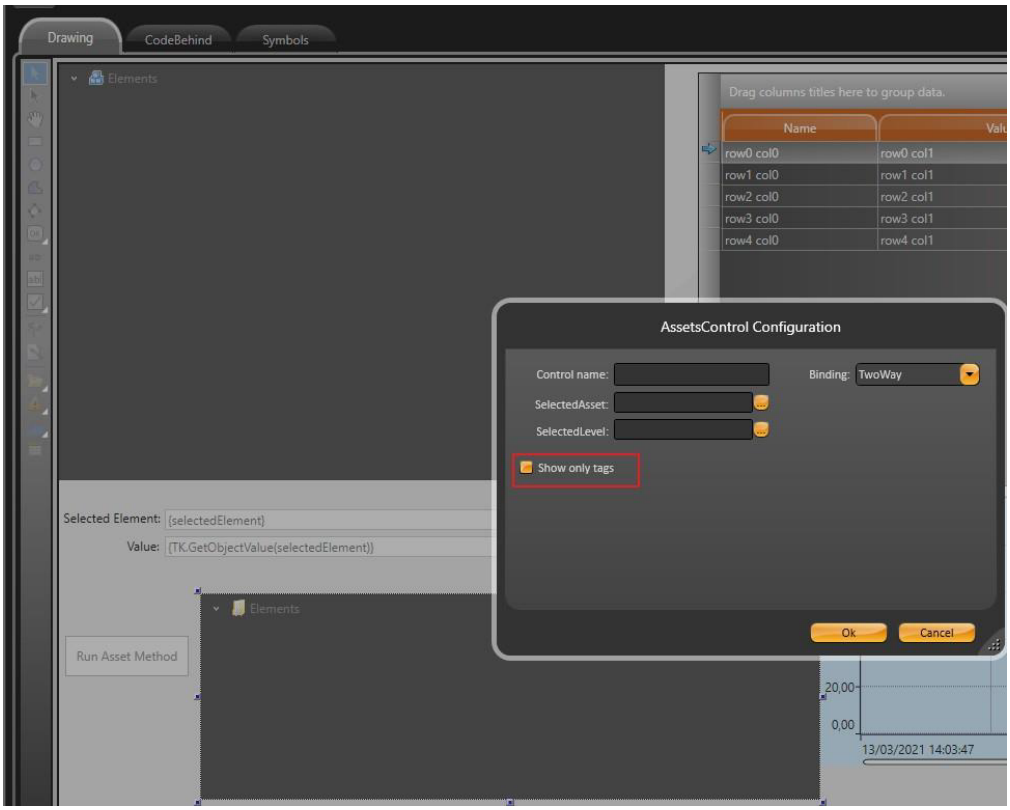
In your project, you can have a full or partial view of the MQTTspB Data Mode. In Edit > Assets, you can create your own Levels and assign MQTT nodes to them, from the Unified Namespace.

This method allows you to import just a piece of the model from the selected node.

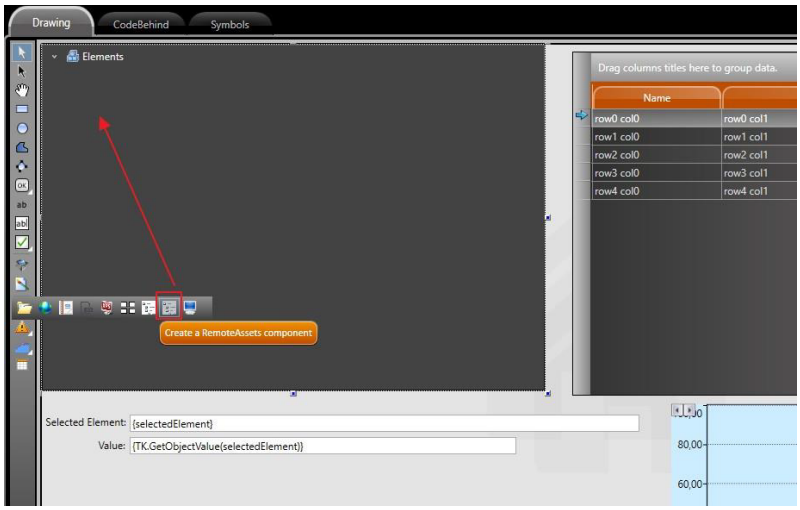
The name of the Level in the Asset Tree (in Runtime) can be edited in the Description column (2).



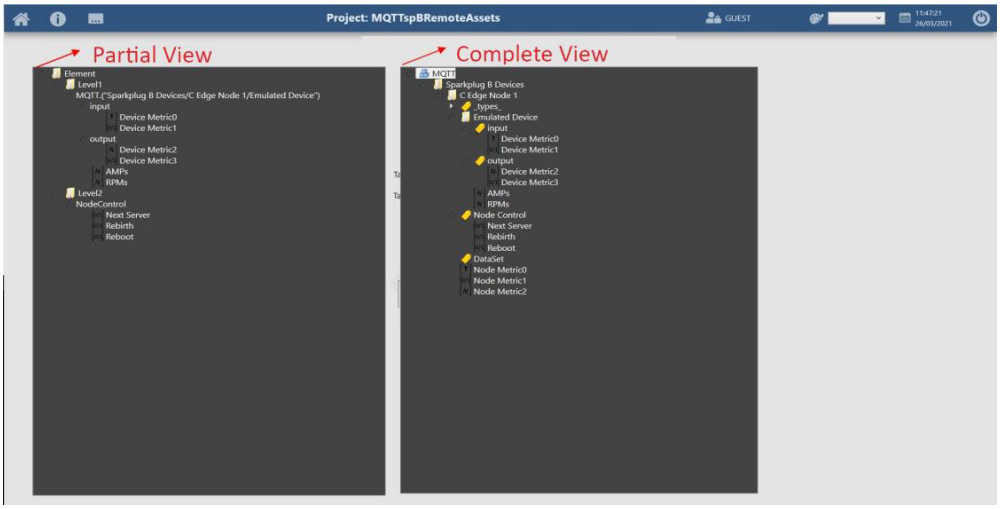
In your Draw Environment, add a component called AssetControl. Open the element's configuration, and uncheck the Show Only Tags CheckBox.



Alternatively, you can import the entire Data Structure by using the RemoteAssets component.



If everything was done correctly, your Asset View should look something like this in Runtime.



It is very important to ensure that the information displayed, both in Runtime and in the Engineering Environment, are Dynamic. The displayed asset tree depends on what information is being sent to the Broker.