Skill MQTT Integration

Router skill for MQTT integration. Choose between TagProvider (dynamic discovery) and Device Module (static tags with alarms/historian) based on what the user needs.

FrameworX offers two paths for connecting to MQTT brokers, plus an interactive browsing tool inside Designer for first-time exploration. This skill helps you pick the right one before loading detailed implementation steps.

Decision Tree

```
User asks about MQTT
  |
  +-- "Browse what the broker has" or "explore MQTT topics"
  |     --> Use Designer's Data Explorer: navigate to DataExplorer.MQTTTools,
  |         start broker/simulator, connect and browse manually.
  |         Then decide TagProvider or Device Module for the actual integration.
  |
  +-- "Connect to MQTT for monitoring" or "dynamic tags" or "asset navigation"
  |     --> Skill MQTT TagProvider
  |     Use when: data is dynamic/unknown upfront, monitoring/visualization focus,
  |     no need for alarms or historian on individual points
  |
  +-- "Create tags from MQTT" or "alarms on MQTT data" or "historian for MQTT"
  |     --> Skill MQTT Device Module
  |     Use when: need local persistent tags, alarms, historian, scripting,
  |     or structured UserTypes (mandatory for SparkplugB)
  |
  +-- "Both browsing AND alarms on critical points"
        --> Hybrid: TagProvider for full namespace browsing +
            Device Module for the specific points that need alarms/historian
        --> Load both skills
```

Two Paths Compared

Aspect	TagProvider (Discovery Services)	Device Module (Channel/Node/Points)
Skill page	Skill MQTT TagProvider	Skill MQTT Device Module
Creates	UnsTagProviders + UnsAssetTree	DevicesChannels + DevicesNodes + DevicesPoints + UnsTags (+ UnsUserTypes for SpB)
Tags	Dynamic (exist only at runtime)	Local persistent tags in UNS
Alarms	Not possible on dynamic tags	Yes, on local tags
Historian	Not possible on dynamic tags	Yes, on local tags
SparkplugB	Works (use MQTTspB protocol)	Works, but REQUIRES UserTypes
Discovery	Full namespace auto-discovered	Must know topic structure upfront
Best for	Monitoring dashboards, asset navigation	Control systems, alarmed points, historical trending

How to Load Sub-Skills

```
search_docs('Skill MQTT TagProvider', labels='skill')
search_docs('Skill MQTT Device Module', labels='skill')
```

Protocol Quick Reference

Both TagProvider and Device Module use the same MQTT protocols. Always call `list_protocols` before writing protocol-dependent tables.

Protocol	Name in FrameworX	Interface	TagProvider	Device Module
MQTT (flat)	`MQTT`	`Custom`	Yes	Yes — simple tags
SparkplugB Collector	`MQTTspB`	`TCPIP`	Yes	Yes — REQUIRES UserTypes
SparkplugB Publisher	`MQTTspBPub`	`TCPIP`	No	Yes — for edge publishing

Built-in Testing Tools

Both paths can be tested with the built-in broker and simulator:

```
designer_action('navigate', 'DataExplorer.MQTTTools')
designer_action('builtin_broker', 'start')   # Starts localhost:1883
designer_action('simulator', 'start')         # Publishes SparkplugB sample data
```

The simulator publishes SparkplugB data: GroupID="Cities", 4 NodeIDs (Barcelona, Bilbao, Madrid, Sevilha), 4 Panels per city, 8 variables per panel (Current, Latitude, Longitude, Name, Power, State, Temperature, Voltage).

See also

- Skill MQTT TagProvider. Dynamic discovery path: TagProvider + AssetTree + Asset() bindings for monitoring and visualization.
- Skill MQTT Device Module. Static-tag path: Channel/Node/Points pipeline with alarms, historian, and scripting on MQTT data.
- Skill Discovery Services. Conceptual foundation for the three connection patterns (Device Module, DataLink, Dynamic Tags).

In this section...