InfluxDB Database Service

Information on how to set up a TagProvider connection with InfluxDB Database.

Reference Modules UNS UI Services InfluxDB

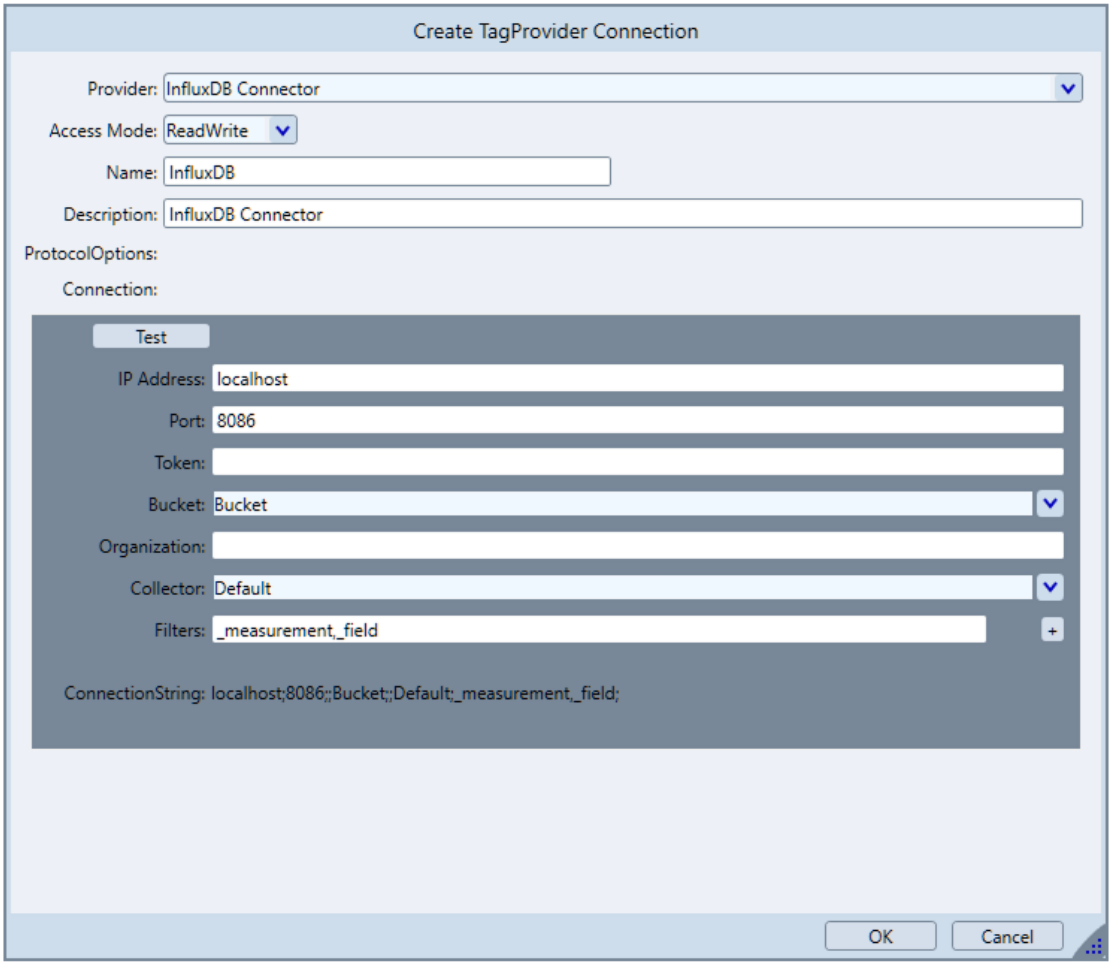
The InfluxDB TagProvider allow dynamic retrieval of data from InfluxDB database, as well to save tags values to the InfluxDB databases. It can be used in read-only mode mode, or updating values, using as an Unified Namespace TagProvider, or it can also be used in Historian, Storage Locations, to act as one of locations where the Historian Module can save its data.

Configuration
- System Requirements
- Connection Parameters

Collectors
- Default Collector
- MQTT Collector
- SolutionHistorian Collector

Configuration

To set up a new TagProvider connection navigate to **Unified Namespace/TagProvider Connections** and press the add button to open the New TagProvider Connection dialog. You can also create your TagProvider Connection directly from the AssetTree. To do this, navigate to **Unified Namespace/Assets Tree** and press  to open the New TagProvider Connection dialog. In the configuration dialog, select the **InfluxDB Connector** on the Provider field and insert the required connection parameters. The same configuration dialog will be presented when selecting **Influx DB** from **Historian/StorageLocations**. After that, press the Ok button to finish the provider configuration. Creating the connection adds no folder to the Asset Tree. To use its data with Asset(), open **Unified Namespace / Assets Tree**, select a folder (create one first if needed) and click **Insert data from TagProvider** on the toolbar: the folder linked to the TagProvider is created inside the selected folder. A linked folder cannot sit at the root of the Asset Tree.



C

System Requirements

This TagProvider requires the user to have access to an InfluxDB Server with support for **API Version 2.0** or greater.

Connection Parameters

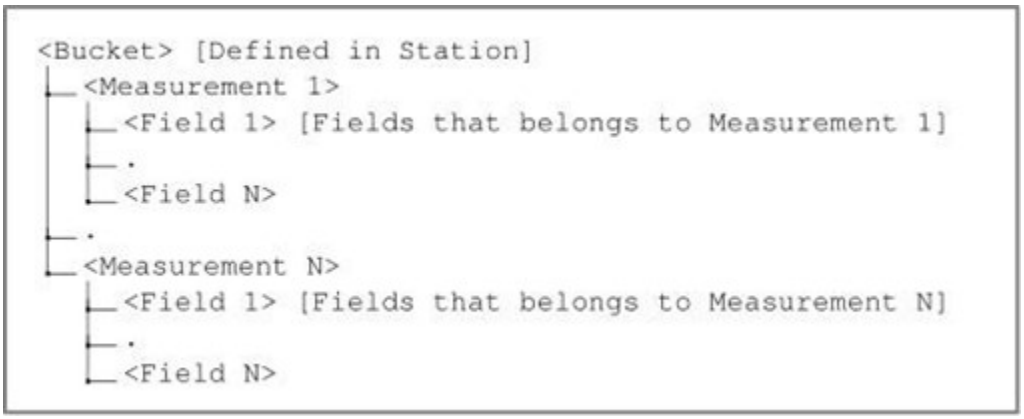
Field	Description
Provider	Specifies the system the TagProvider will interface with. In this case, it interfaces with the InfluxDB Connector.
Access Mode	Determines permissions for interacting with tags. **Read** allows only data reading. **Write** permits only data writing. **ReadWrite** enables both reading and writing.
Name	User-defined name for the TagProvider instance. Identifies the specific configuration in the system.
Description	Defines a short description for the TagProvider's purpose or the systems it interfaces with. Provides context for its usage.
ProtocolOptions	Defines specific communication settings for the connection to the device.
Connection	Defines connection settings for the InfluxDB, including IP address, port, token, bucket, and organization.
IP Address	Specifies the IP address of the InfluxDB server.
Port	Sets the port number used for communication with the InfluxDB server.
Token	Specifies the token for authentication with the InfluxDB server.
Bucket	Defines the bucket within InfluxDB to retrieve or store data.
Organization	Specifies the organization associated with the InfluxDB instance.
Test	Verifies the connection settings to ensure successful communication with the specified server.
ConnectionString	Displays the full connection string used to connect to the InfluxDB server. Provides a summary of all connection parameters.

- **Collector**: Specifies the type of Collector to be used. Each one have a specific behavior in Runtime. Options are:
  - *Default*: Connects to any InfluxDB Database without any custom TreeView is based only on **Filter** configuration.
  - *SolutionHistorian*: Is used to **Write** ProjectTags Historian data in a custom.
    - **Filters** field will **always** be disabled, with **measurement** and **field** columns added to.
    - When Provider is used as **Historian**, Historian TableName will be used as **measurement**.
    - TreeView structure will behave different than **Default** and **MQTT** More details available in following sections.
  - *MQTT*: Is used to Read Data that was stored using Telegraf's MQTT Plugin pre-defined column structure.
    - The **Topic** information **MUST** be stored in a column named **topic**.
    - **Filters** field will **always** be disabled, with **measurement**, **topic** and **field** columns added to
    - TreeView structure will behave different than **Default** and **tProj Historian**. More details available in following sections.
- **Filters**: Defines a hierarchical order for columns, separated by **comma** (","), when assembling the TreeView.

1. You can browse the columns to be discarded at the "+"
2. Click on + or - to add / remove And *Up/Down* arrows to change the filter order.

> **Collector Selection**
>
> When using the InfluxDB in conjunction with Unified Namespace TagProvider, the Collector Default and MQTT can be used.
>
> When using the InfluxDB as a Historian Storage Location, the Collection SolutionHistorian shall be used.

The Filters are used to define the Query to be executed when assembling the TreeView. In this example with **measurement** and **field**, the structure would be as follows:



CustomColumns in InfluxDB

TagProvider connections **cannot** create new CustomColumns in InfluxDB Database, only access existing ones. To create new CustomColumns, use the Device Module Configuration.

Collectors

Default Collector

This collector will access any InfluxDB Database, and create the TreeView Structure based on the Filters.

This Collector can **Read/Write** data, and it is used with the Unified Namespace TagProviders.

MQTT Collector

This Collector is used to Read Data from an InfluxDB Database Structure that was stored using Telegraf's MQTT Plugin pre-defined schema, with MQTT topic stored in a column called **topic**.

The filters are, by default, defined as **_measurement**, **topic** and **_field**.

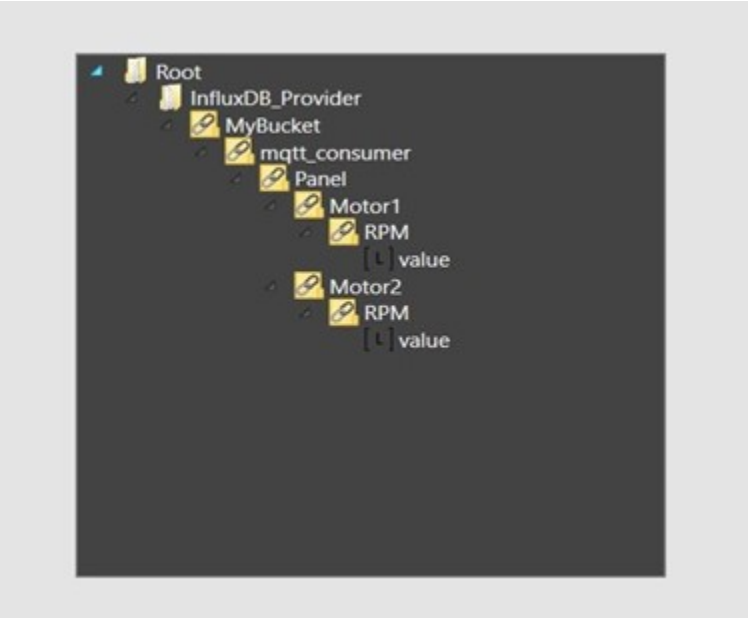
In Runtime, the TreeView Structure will use the MQTT Topic similar to what is done in MQTTspB ExternalTags.

As an example, let us assume we have an MQTT Client publishing to Topics:

```
Panel/Motor1/RPM
Panel/Motor2/RPM
```

The **_measurement** is defined as **mqtt_consumer** and **_field** is **value**.

Our TreeView Structure is illustrated below.



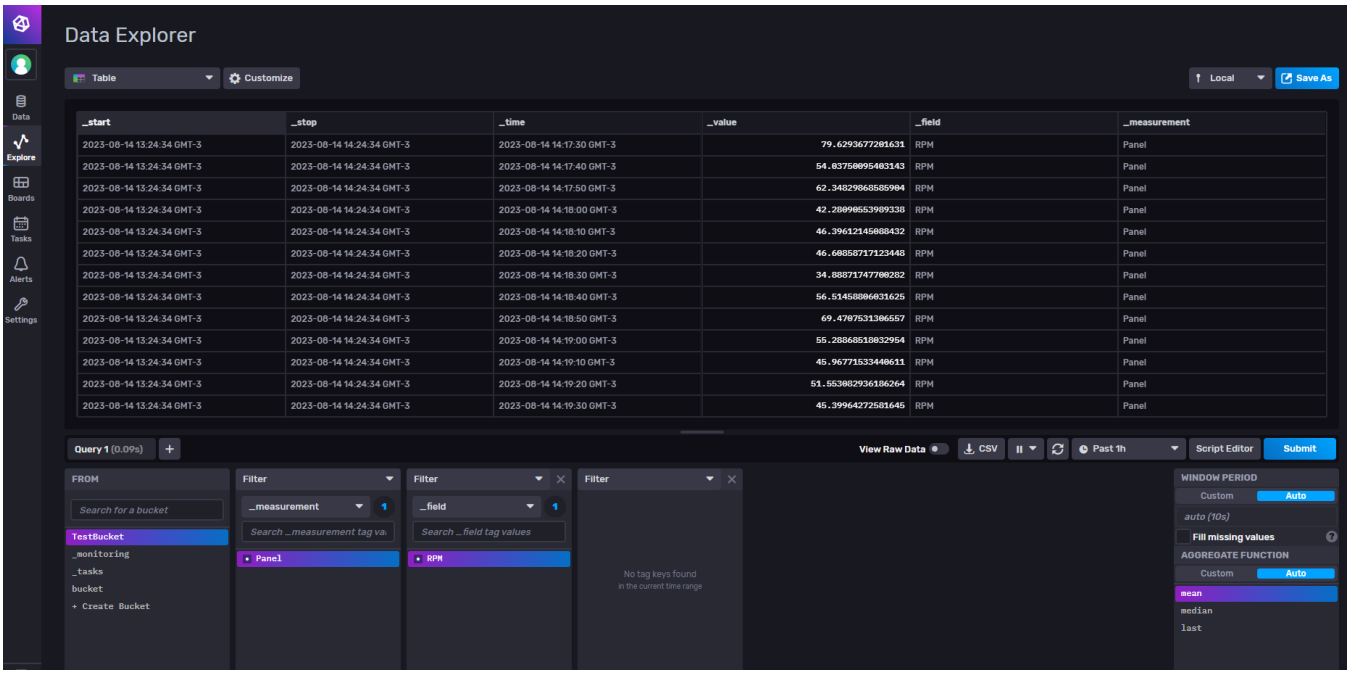
SolutionHistorian Collector

This collector is used with the Historian Storage Location configuration.

In this case, the InfluxDB is used as one of the repositories for the Historian Module of the solution.

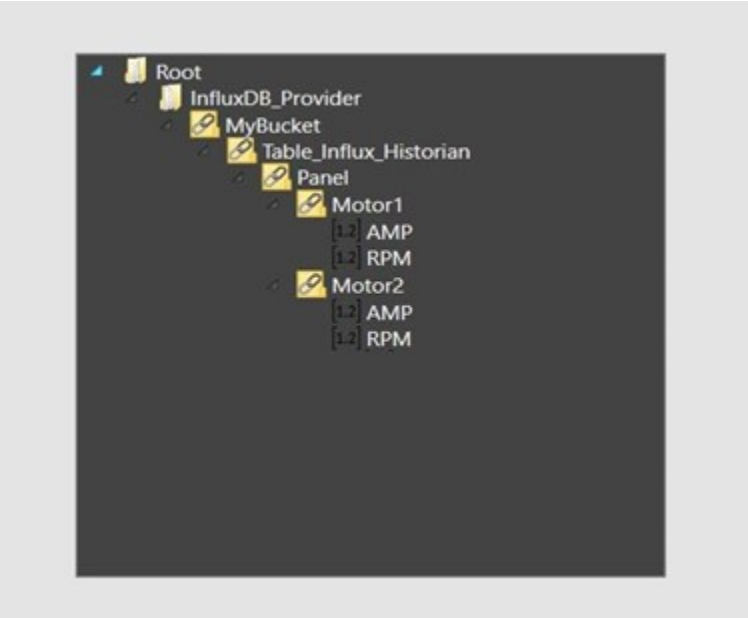
In this Configuration, Tags added on **Historian / Historian Tags**, into a Table targeted for InfluxDB Provider, will be automatically logged in your Database.

The **Historian Table** Name will act as our **_measurement** and the Complete TagName will be added to the **field** column.



Reading Historical Data

By using the Complete TagName as **_field**, the TreeView can use that information to transform that name into a multi-level structure as illustrated below.



In this section...
