



Kymera Influx DB Historian Module

User Guide

Table of Contents

Introduction	2
Setup	3
Usage	6
Scripting	9
Patch Notes	11
Support	11

Introduction

The Kymera Influx DB Historian module allows Ignition to store tag history to an Influx database.

The module provides Ignition users a purpose-built time series database, and offers better performance compared to traditional storage platforms because of InfluxDB's ability to handle high write and query loads.

Setup

This module requires a running InfluxDB instance, accessible to the Ignition server. The module can connect to either Influx v1.8+ or v2.x by setting the 1.8+ compatibility setting. If connecting to a 1.8+ instance, Flux must be enabled on the Influx server for the module to connect to it.

The download for InfluxDB, as well as further documentation on how to set up the server, can be found at <https://www.influxdata.com>.

After the module is installed, a new historical provider will be available under the **Tags>History** menu in the Configure section.

Click on “Create a new Historical Tag Provider”.

Select the Kymera InfluxDB provider and click “Next”. Most of the initial settings for a new historical tag provider are prepopulated, but they can be modified. By and large, the default settings will be sufficient, but the connection settings will need to be filled out. The following properties are available at creation:

Connection Settings

Provider Name	The name of the provider. It’s best if a unique one is chosen.
Influx Server Address	The full URL of the Influx server including the port. Default is <code>http://localhost:8086</code> .
Token/Password	The API token for authentication if using v2. If v1.8+ compatibility is enabled, then this is the password field.
Organization name/Username	The organization name if using v2. If v1.8+ compatibility is enabled, then this is used for the username.
Bucket name/Database	The bucket name if using v2. If v1.8+ compatibility is enabled, then this is used for the database name.

Compatibility Settings

Enable v1.8+	Set whether or not the driver is connecting to an Influx v1.8+ endpoint, default false.
v1.8x Retention Policy	Retention policy if v1.8+ compatibility enabled, default autogen.

Write Settings

Flush Amount	Number of data points to collect in a batch, default 1000.
Flush Duration	Number of milliseconds before batch is written, default 1000.

Jitter Amount	Number of milliseconds to increase batch flush interval by a random amount, default 0.
Retry Interval	Number of milliseconds to retry unsuccessful write, default 5000.
Max Retries	Number of max retries when write fails, default 5.
Max Retry Delay	Max delay between each retry attempt in milliseconds, default 125,000.
Max Retry Time	Max total retry timeout in milliseconds, 100,000.
Exponential Base	Base for exponential retry delay, default 2.
Buffer Limit	The max number of unwritten stored points, default 10000.

Historian Settings

Tagpath Active Days	Number of days a tag is considered active when building a tag cache, default 30.
Result Frequency	Expected result density on the average query, default 1000.
Max Query Results	Max number of query results to return, default 300.
Query Buffer Seconds	Width in seconds to pad query in order to find edges, default 60.
Query Buffer Window	Width in windows to pad query in order to find edges, default 5.
Compress Quality	Option to compress the tag quality data. Initial tag quality is stored, but unless a quality change occurs, the quality column remains blank. If set to true, storage performance improves. Default false.
Enable Tag Rebuild	Sets whether the local tag cache tracking tag datatypes is rebuilt on the historian or module restarting. Default false.

Store and Forward Settings

Data Tape Path	Relative path of where data tape store is to be held. Default is data.
Quarantine Tape Path	Relative path of where quarantine tape store is to be held. Default is quarantine.

Max Tape Records	Max number of records to be held in the tape store, default 10000.
Max Memory Records	Max number of records to be held in the memory store, default 10000.
Max Memory Forward Time	Max amount of milliseconds a record is allowed to spend in memory, default 10000.
Max Memory Forward Amount	Max number of records allowed in memory at a time, default 25000.
Max Disk Forward Time	Max amount of milliseconds a record is allowed to stay on the tape, default 60000.
Max Disk Forward Amount	Max amount of records to forward each time, default 10000.

Tape Store Overflow Warning Criteria Settings

Max Allowed Tape Store Delay	The max amount of milliseconds records are allowed to spend in the tape store before a warning is logged in the gateway. This value is calculated as an aggregate using Poll Point Count to determine the window size. Default 7,200,000 or 2 hours.
Pause Between Checks	Time in milliseconds to wait between polling the tape store, default 900,000 or 15 minutes
Poll Point Count	The amount of records used to calculate the tape store delay. System will wait to have this many points before calculating running average.

Finalize the creation of the provider by clicking the “Create New Historical Provider” button at the bottom.

Usage

After creating the tag provider, it can be selected like any other history provider in the Designer. Double click on the tag, or right click on the tag and select "Edit tag".

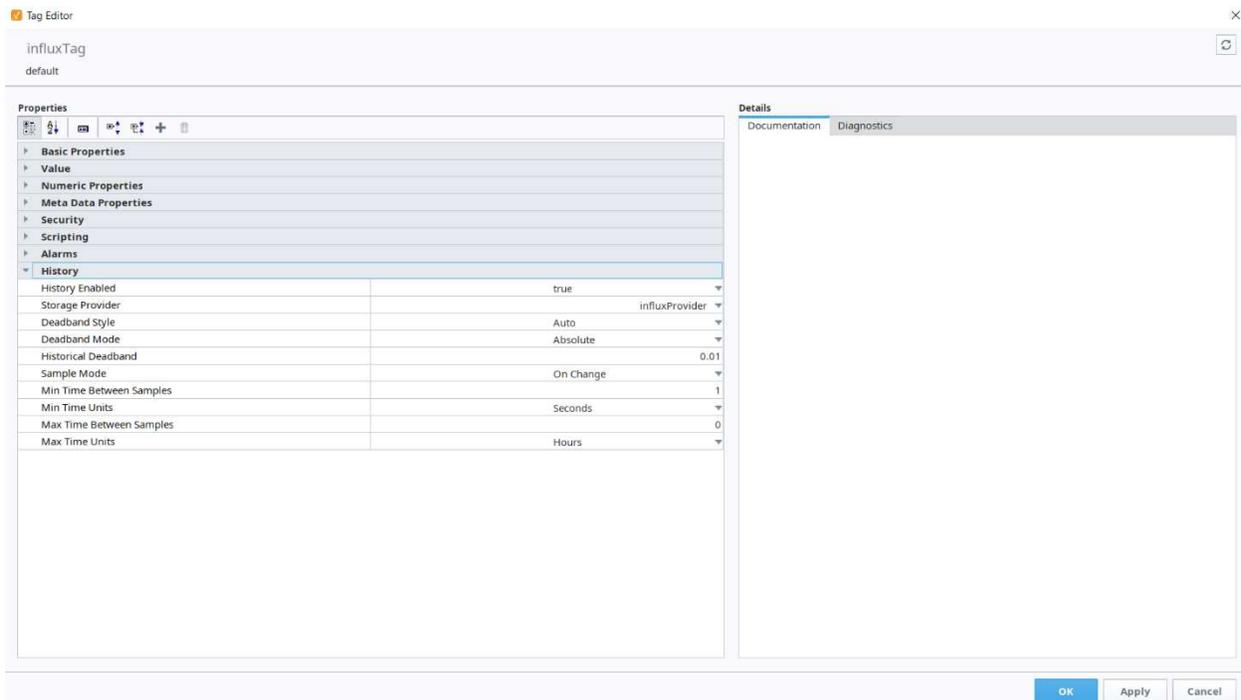
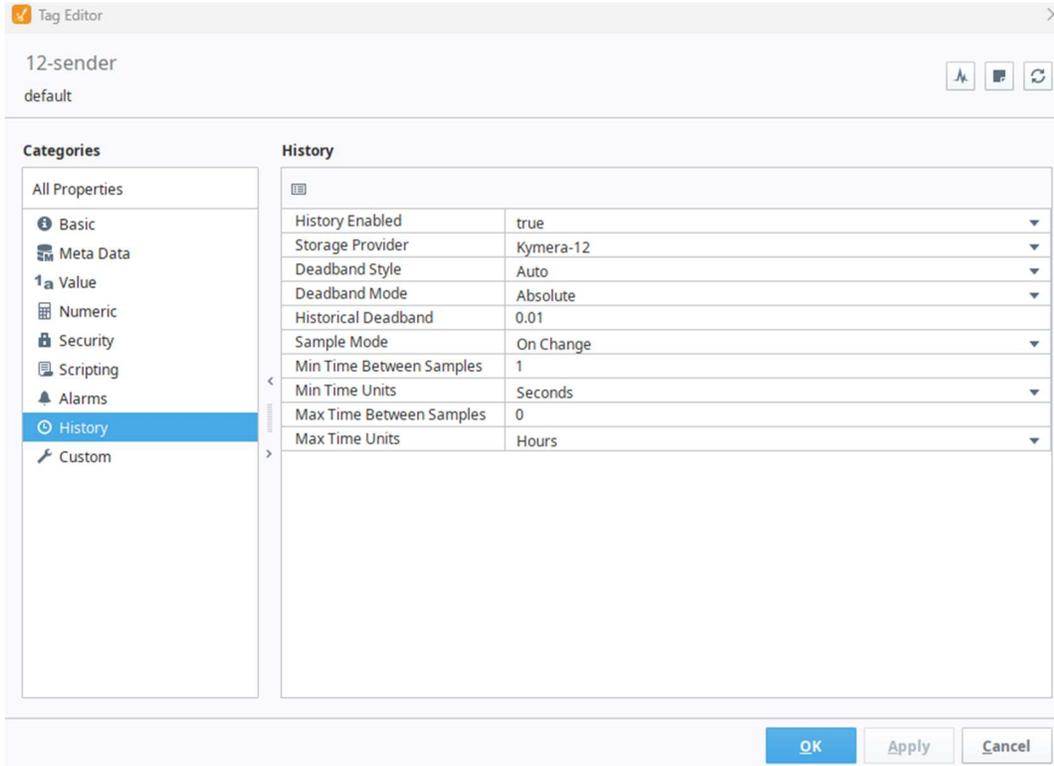


Fig. 1: Setting up tag history in Ignition 7.9 and 8.1

After recording any historical tag data, the tag data will be visible Easy Chart components...

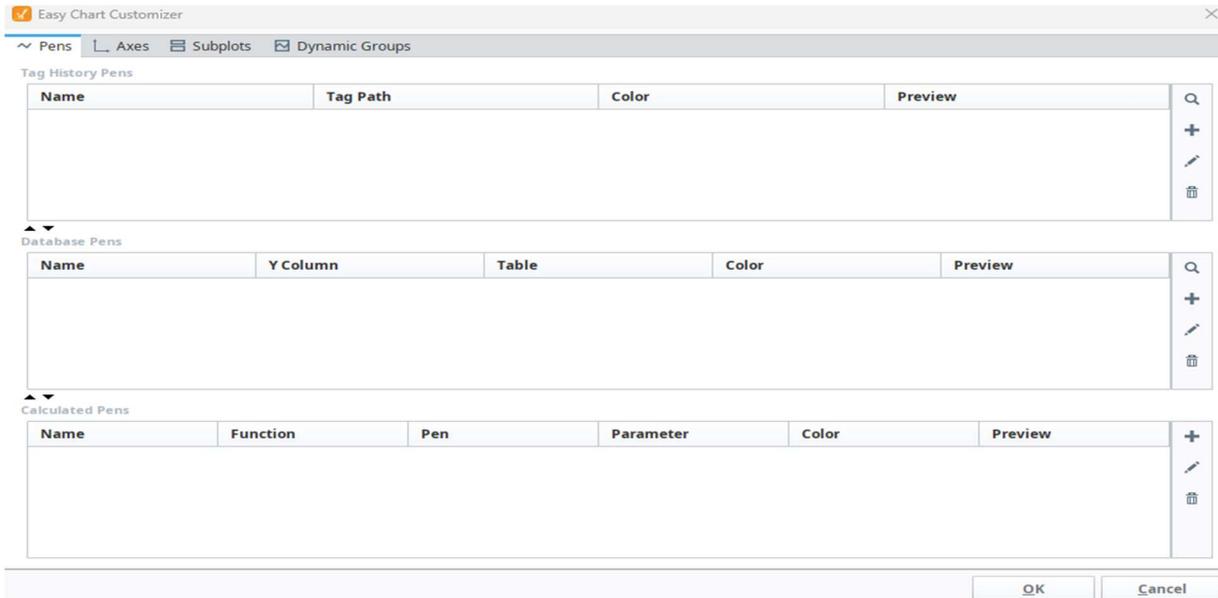
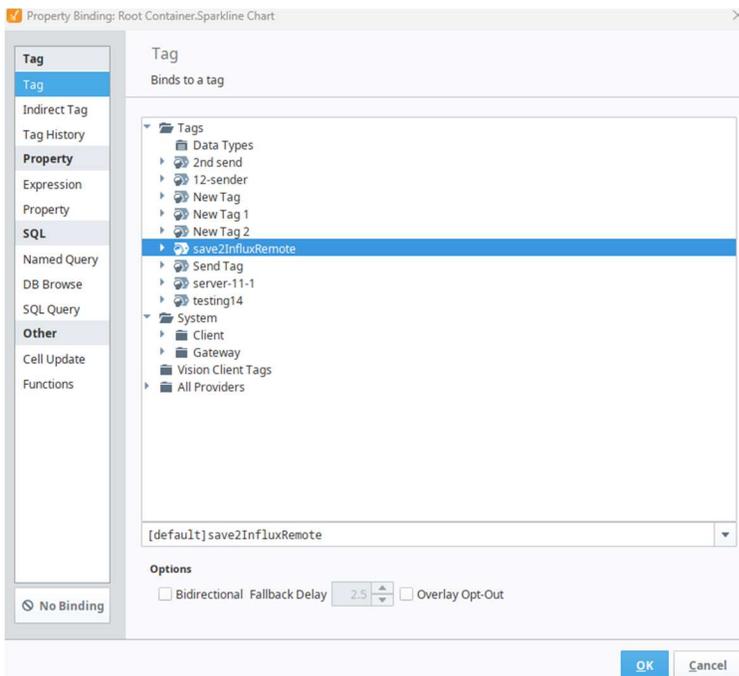


Fig. 2: Browsing Influx tag historian using Easy Chart

as well as any components or properties that use tag history bindings, like Sparkline Charts or tables.



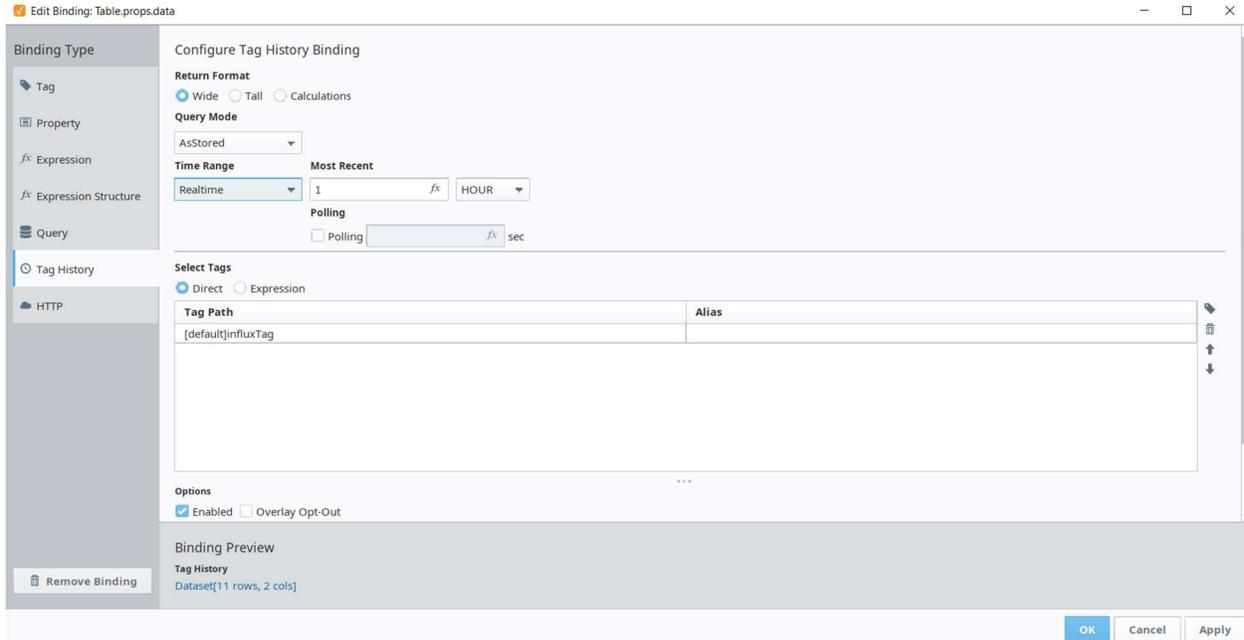


Fig. 3: Browsing Influx tag historian using tag history binding in Ignition 7.9 and 8.1

Scripting

The following scripting functions are available to interact with the Influx database.

system.influx.insert()

Description

Inserts data into an Influx database.

Syntax

```
system.influx.insert(historianName, database, measurement, tagHeader, tagValues,  
fieldHeader, fieldValues, timestamps)
```

Parameters

str historianName – Name of a preestablished Influx connection on the gateway. Used for credentials and retention policy.

str database – Name of a preexisting database in the Influx database/bucket.

str measurement – Name of the measurement to store the data to.

str[] tagHeader – List of the tag keys to be added to each data point.

str[][] tagValues – List of lists of the tag values to be associated with the datapoints.

str[] fieldHeader – List of the field keys to be added to each data point.

str[][] fieldValues – List of lists of the field values to be associated with the datapoints.

long[] timestamps – List of the timestamps associated with the datapoints, in milliseconds.

Returns

None

Scope

Client, Designer, Gateway

Example

```
historian = "influx" # this tag historian exists on the gateway already  
database = "test" # this Influx database must also exist  
measurement = "measurement_name"  
tagHeader = ["tag_odd", "tag_even"]  
tagVals = [["tag1", "tag2"], ["tag3", "tag4"]]  
now = system.date.toMillis(system.date.now())  
timestamps = [now, now]  
fieldHeader = ["field1", "field2", "field3"]  
fieldVals = [[1, False, "hello"], [2, True, "world"]]  
system.influx.insert(historian, database, measurement, tagHeader, tagVals, fieldHeader, fieldVals,  
timestamps)
```

system.influx.query()

Description

Execute a general Flux statement using the credentials in the specified historian.

Syntax

```
system.influx.query(historianName, flux)
```

Parameters

str historianName – Name of the Influx provider to use for credentials. If the given name is not a match to a connection on the gateway, it will take the first Influx connection available in the gateway.

str flux – The Flux query to execute against the Influx database. If the historian is v1.8+, then the Influx endpoint must have Flux enabled.

Returns

If the Flux string is a query for data, will return a List of List of Maps. The first level of Lists represents the tables, while the second represents the individual records. The Maps represent the whole record. Keys to the Map will depend on the query and bucket. If the Flux statement isn't a query, returns null.

Scope

Client, Designer, Gateway

Example

```
# querying our own records, which has tagPath, _time, _field, _measurement and _value as record
# keys

historian = "influx"
query = "" from(bucket: "kymera")
  |> range(start: 2022-05-25T10:00:00Z, stop: 2022-05-26T10:00:00Z)
  |> filter(fn: (r) => r["_measurement"] == "ignition-server.default")
  |> filter(fn: (r) => r["tagPath"] == "integerTag")
  |> filter(fn: (r) => r["_field"] == "iVal")""

tables = system.influx.query(historian, query) # returns list of table(s)

table = tables[0] # gets first table
record = table[0] # gets first record
tagPath = record["tagPath"] # gets the tagpath
```

Patch Notes

Sept 2023 build : Added system.influx.insert compatibility for Influx 2.x. Added auth check for Influx 2.x. Disabled tag cache rebuilding on module restart and added gateway setting to enable/disable. Added tape store delay feature to monitor amount of time records stay in tape store. Fixed bug where tag cache was being rebuilt too often when browsing available tags in historian. Added ability to query live value. Added raw and natural query aggregation. Added fix for remote tags without system names. Implemented 8.1.32 tag historian API. Fixed null pointer error when Max Disk Forward Time is lowered.

Support

1 year of technical support during regular business hours - Monday - Friday 8AM to 4PM MST.

- Free Upgrades with Ignition updates (I.E. 8.0 to 8.01).
- Support may be contacted via email, at support@kymerasystems.com, or via phone, at 1-800-470-2302. Please allow up to 24 hours for a response from our support team.