



Kymera Influx DB Historian Module

User Guide

Table of Contents

Introduction	2
8.3 Update Procedure	3
Store and Forward	4
Setup	5
Usage	9
Scripting	12
Patch Notes	16
Support	16

Introduction

The Kymera Influx DB Historian module allows Ignition to store tag history to an Influx v1, v2 or v3 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.

8.3 Update Procedure

Before proceeding to upgrade the Kymera InfluxDB module, please ensure that the module currently installed is version 1.11.43.2, which is compatible with Ignition versions 8.1.43 to 8.1.45. Before updating, take a gateway backup of the system and follow Inductive Automation's guidelines on upgrading to 8.3.

If you cannot upgrade to this version or have other questions about your upgrade, please reach out to our support team at support@kymerasystems.com.

After upgrading the Ignition version, navigate to the modules page, under **Platform -> System -> Modules**. If the 8.1 version of the Kymera Influx module is still present, uninstall it. This will require a server restart.

Once the server has restarted, install the 8.3 version of the module and restart the server once again. After installation is finished, confirm that the Influx connection settings are correct under the **Services -> Historians** link and verify tags are historizing.

New to the 8.3 version is the introduction of 5 new fields, Enable Measurement Override, Measurement Name Override, Enable Provider and Server Tag Overrides, Influx Provider Tag Key and Influx Server Tag Key. Our Influx module will ordinarily store tag history data in an Influx measurement/database as a combo key of the server name and the tag provider, e.g. server1.default. The Measurement Name Override field will allow users to change it to be something else, as long as the Enable Measurement Override field is checked. Similarly, the Server Tag Override and Influx Provider Tag Override fields will write 2 additional tags for each record, as long as the Enable Provider and Server Tag Overrides checkbox is checked. The default values will store the current tag provider and server name, e.g. provider=default, server=server1, or something else if desired based on the entry in the fields. **Please note that enabling these fields will increase your database's cardinality, which can impact performance. Please also note that enabling the measurement override may result the data losing the original references needed to query for historical data through the Ignition historian. Please proceed with caution before exploring these options.**

If migrating from an 8.1.X build with these overrides, if the measurement override field is blank or the default value of {serverName}.{providerName}, the enable measurement name override field will start as disabled. If either the server or provider field is blank or the default values of {providerName} or {serverName}, the enable provider and server tag override field will begin as disabled.

Store and Forward

The Kymera Influx module uses a similar store and forward system as Ignition does. When the Influx module receives historical data, it is converted to line protocol, further information on line protocol can be found here: <https://docs.influxdata.com/influxdb/v2/reference/syntax/line-protocol/?section=influxdb%252Fv2%252Freference>

Our line protocol format is built to store the server name, the tag provider, the value, the tag quality, the tag name, and the timestamp. This format is designed to reconstruct the tag's historical information when queried back from an Influx database. The format it follows is:

`$serverName.$tagProviderName, tagPath=$fullTagPath (iVal/fVal/bVal/sVal)=$value, quality=$tagQuality $timestamp`

e.g. `server1.default,tagPath=path/to/tag iVal=123456789i,quality=192i 1770912752451`

Each of these changes is grouped into something called a `LineProtocolPage`. This represents a batch of tag history to be sent to the Influx database. This page will wait for a preset amount of time (default 10000ms) or until the max number of records have been added to a page (default 5000) before the page is added to a queue to be forwarded to the Influx database. At that point, a new page is started and historical data is added to the new page.

When a protocol page is not forwarded to the Influx database after a configurable amount of time (default 60000ms), it will be cached in the `ignition/data/datacache/nameOfHistorian/dataDir` folder. The files have the `.line` file extension and are in plaintext. Each entry is in line protocol format and can be entered into an Influx database through their CLI or web UI.

If after some more time the page is not forwarded (default 300000ms), the page will be unloaded from memory and stay in the `dataDir` directory until the module is able to catch up with forwarding pages, at which point the pages will be reloaded into memory and cleared off disk. The max number of pages that can be stored in this way is 10000 by default but can be configured in the gateway. If the number of pages exceed 10000, then the oldest ones will be deleted until the number of pages is once again 10000. If the connection to the Influx database is down for example, depending on the rate of tag history data being generated, many files may need to be cached on the Ignition server, so be mindful when setting the file limit and choosing hardware.

If during the sending process an exception occurs, the page is requeued up to 5 times to be sent again. If it still fails to send, it is instead stored in the `ignition/data/datacache/nameOfHistorian/quarantineDir` folder. This data cannot be accessed through the Ignition web UI, and can only be accessed by navigating to the `quarantineDir` folder. The page data will still be in line protocol format.

Please note that if the `dataDir` and `quarantineDir` folders are deleted, there is a risk of losing historical data. Please do not proceed before contacting support.

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. This version of the module also supports connecting to InfluxDB v3 Cloud and InfluxDB v3 Core. **Please note that v3 Core has a 72 hour query window limitation.**

If connecting to a 1.8+ instance, Flux must be enabled on the Influx server for the module to connect to it.

If connecting to a InfluxDB v3 instance, an additional Java wrapper argument must be added to the ignition.conf file. This setting is "--add-opens=java.base/java.nio=ALL-UNNAMED" and will require the gateway to be restarted before the changes can be applied so the v3 connections can work.

The download for InfluxDB, as well as further documentation on how to set up the server, can be found at <https://www.influxdata.com>. Before the module can connect to the database, please provision the Influx endpoint with a bucket/database and a token API.

After the module is installed, a new historical provider will be available under the **Services->Historian** section.

Click on "Create Historian".

Select the Kymera InfluxDB v1/v2 provider or Influx v3 provider and click "Next". **For this current release, Ignition's webform does not support prepopulated defaults or required field enforcement, so all fields will need to be filled in by hand. The intended defaults will be described in the field description.** 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 http://localhost:8086.
Token/Password	The API token for authentication if using v2 or v3. If v1.8+ compatibility is enabled, then this is the password field.
Organization name/Username	The organization name if using v2 or v3. If v1.8+ compatibility is enabled, then this is used for the username.
Bucket name/Database	The bucket name if using v2 or v3. 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. For v1/v2 connections only.

v1.8x Retention Policy

Retention policy if v1.8+ compatibility enabled, default autogen.

Write Settings

Enable Gzip

Enable Gzip compression while sending data to Influx, default false. For v1/v2 connections only.

Historian Settings

Enable Tag Rebuild

Sets whether the local tag cache tracking tag datatypes is rebuilt on the historian or module restarting. Depending on the number of tags, this can be a slow process if enabled. Default false.

Tag Path Activity Days

Number of days a tag is considered active when building a tag cache, default 30. **If using v3 core and enabling tag rebuild, set this value to 3 or less.**

Compress Tag 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.

Query Buffer Seconds

Width in seconds to pad query in order to find edges, default 60.

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 Window

Width in windows to pad query in order to find edges, default 5.

Backfill Seek Length

In the event there is a break in the data, the amount of time to search for a past value. Default is 86400 seconds, or 24 hours.

Disable Null Check Logger

The Influx module rejects null value changes and logs an error message in the gateway. This setting disables the log message. Default false.

Store and Forward Settings

Max Pages

Max number of pages in the page store, and max number of text files representing those pages allowed to be stored on the Ignition server. Each page corresponds to a single instance of forwarding to InfluxDB. If max number of pages is exceeded, the oldest pages are dropped. Default 10000.

Max Page Size

Max number of records that can be in a page. A record corresponds to a single historized datapoint of a tag. Must be in the range of 1-5000, default 5000.

Max Memory Forward Time

Max milliseconds a record can spend in memory, without being written to disk. Default 60000.

Page Stale Time

Max milliseconds a page will wait for records before being added to the forward queue. This is to facilitate grouping of data into a single request. A filled page will automatically advance to forwarding. Default 10000.

Page Unload Time

Max milliseconds the data for a page can be held in memory before the data is unloaded from memory. This value must be greater than the max memory forward time. Default 300000.

Advanced Settings

Enable Measurement Name Override

Enables the measurement name override. CAUTION: Enabling this feature can affect throughput because of increased cardinality. Default false.

Measurement Name Override

Defines the bucket/measurement name used to store data in the Influx database. Default is {serverName}.{providerName}.

Enable Provider and Server Tag Overrides

Enables the provider and server tag override. CAUTION: Enabling this feature can affect

throughput because of increased cardinality.
Default false.

Influx Provider Tag Key

Defines the tag value used to store to the provider tag key in the Influx database. Default is {providerName}.

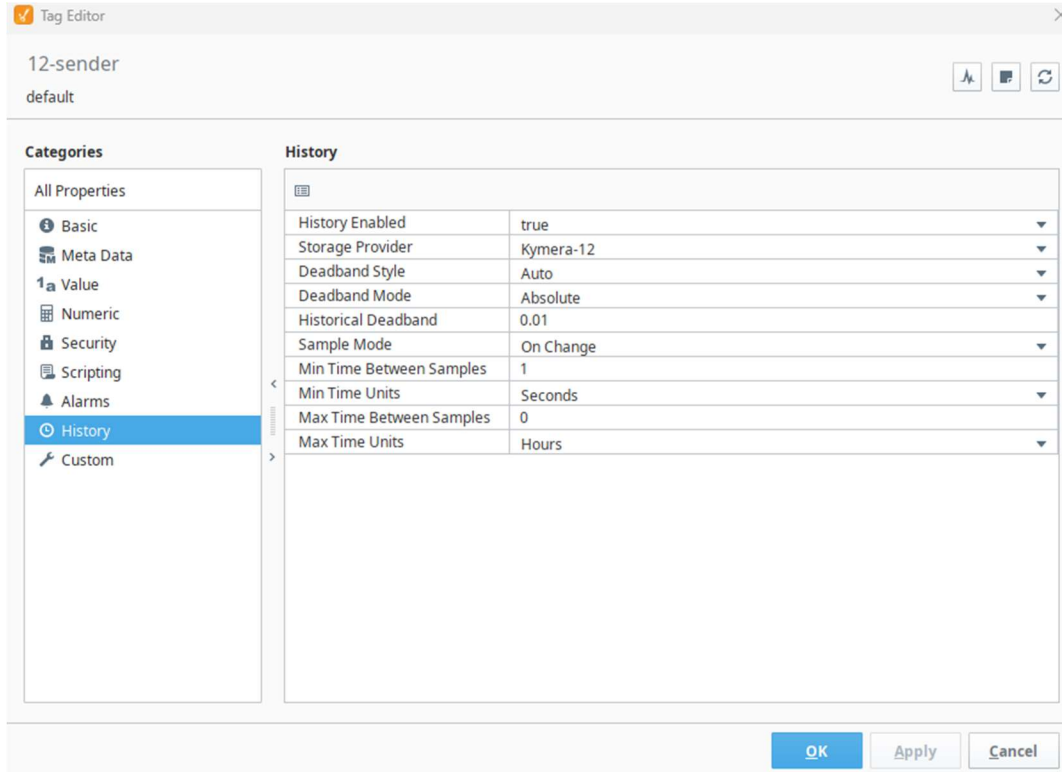
Influx Server Tag Key

Defines the tag value used to store to the server tag key in the Influx database. Default is {serverName}.

Finalize the creation of the provider by clicking the “Create Historian” 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".

The image shows a 'Tag Editor' dialog box for a tag named '12-sender' with a 'default' provider. On the left, a 'Categories' sidebar lists 'All Properties', 'Basic', 'Meta Data', 'Value', 'Numeric', 'Security', 'Scripting', 'Alarms', 'History' (which is selected and highlighted in blue), and 'Custom'. The main area, titled 'History', contains a table of properties. The table has two columns: the property name and its value. The properties are: 'History Enabled' (true), 'Storage Provider' (Kymera-12), 'Deadband Style' (Auto), 'Deadband Mode' (Absolute), 'Historical Deadband' (0.01), 'Sample Mode' (On Change), 'Min Time Between Samples' (1), 'Min Time Units' (Seconds), 'Max Time Between Samples' (0), and 'Max Time Units' (Hours). At the bottom right of the dialog are three buttons: 'OK', 'Apply', and 'Cancel'.

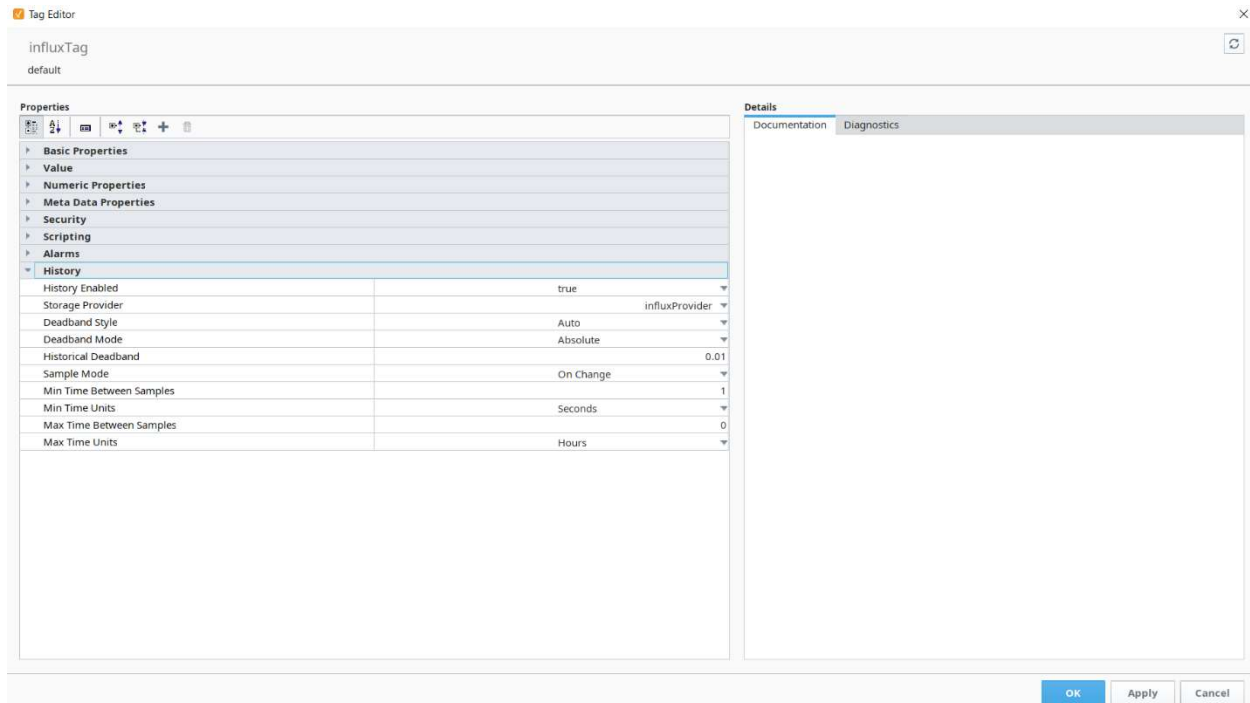


Fig. 1: Setting up tag history in Ignition 7.9 and 8.1

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

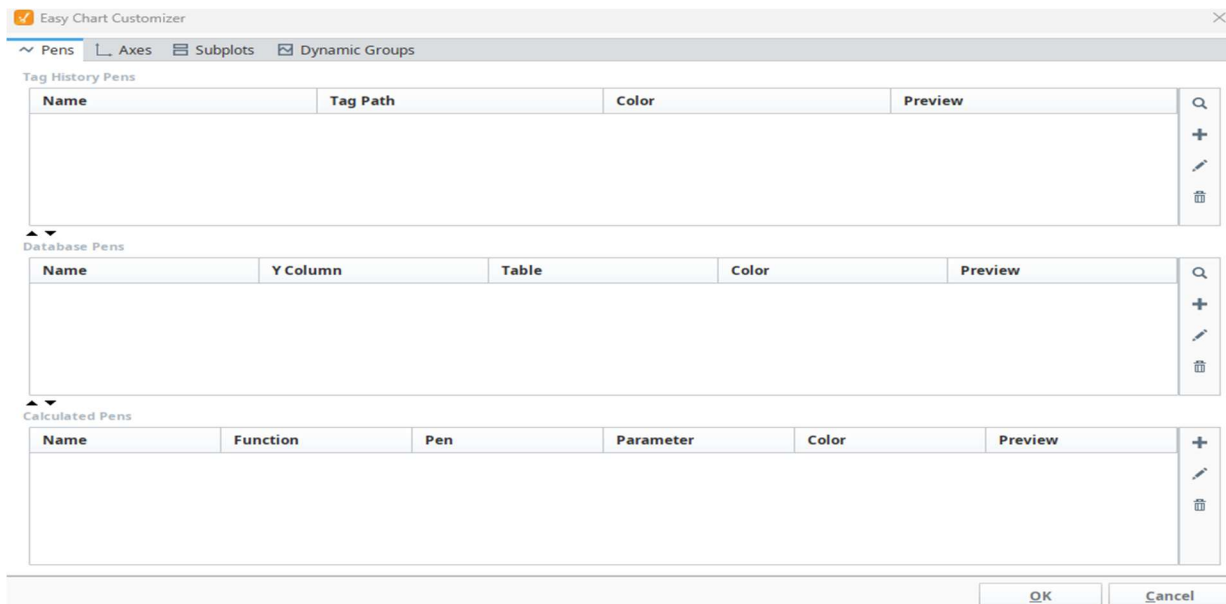


Fig. 2: Browsing Influx tag historian using Easy Chart

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

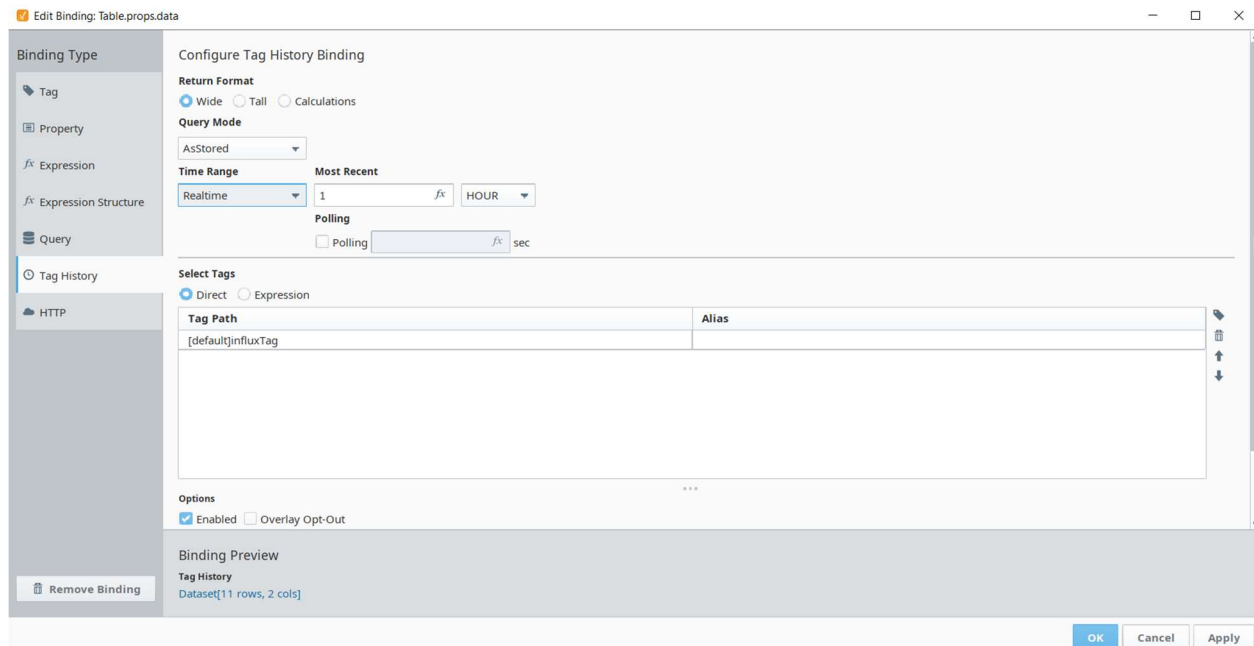
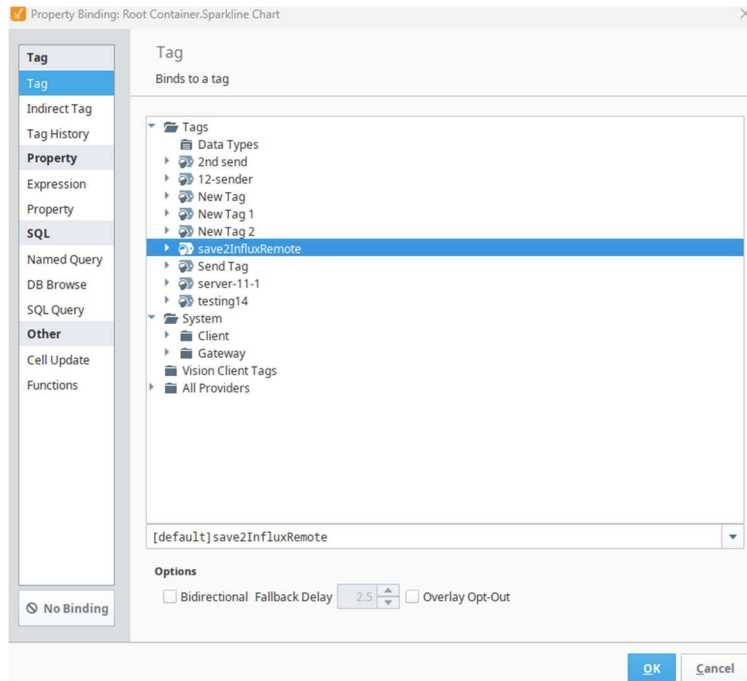


Fig. 3: Browsing Influx tag historian using tag history binding in Ignition 7.9 and 8.1

Scripting

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

system.influx.insert()

Description

Inserts data into an Influx v1, v2 or v3 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.

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

List tagValues – List of lists of the tag values to be associated with the datapoints. Each sublist must have the same number of elements as the tagHeader, e.g. if there are 2 tags, each sublist must have 2 tags.

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

List fieldValues – List of lists of the field values to be associated with the datapoints. Each sublist must have the same number of elements as the fieldHeader, e.g. if there are 2 fields, each sublist must have 2 values.

List 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 for Influx v1 and v2 databases 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
```

system.influx.v3query()

Description

Execute an InfluxQL statement for Influx v3 databases using the credentials in the specified historian.

Syntax

```
system.influx.v3query(historianName, influxql)
```

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 InfluxQL query to execute against the Influx database.

Returns

If the InfluxQL string is a query for data, will return a List of arrays. The length of the array is based on the number of columns in the query statement. If the InfluxQL statement isn't a query, returns null.

Scope

Client, Designer, Gateway

Example

```
# querying our own records, which has time, sVal, bVal, iVal, fVal and quality as field
# keys, and tagPath as a tag

historian = "influxv3"
query = "" select time, "bVal", "quality" from "server1.default" where
"tagPath"='random/randomboolean1' and time >= 17248
13995000ms and time <= 1724817660000ms ""

rows = system.influx.v3query(historian, query) #return list of rows

row = rows.get(0) # gets first row
val = row[2] # gets value
measurement = row[0] # Influx v3 includes the measurement name in every row
```

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.

Dec 2023 build: Changed boolean queries to only support the last value aggregation because Influx does not support other aggregation types.

Jan 2025: Changed from asynchronous writes to Influx to synchronous writes to better support store and forward behaviour. Changed from tape store usage to line protocol based page stores. Removed a number of older settings associated with Ignition's store and forward pipeline and instead replaced with our own store and forward pipeline for better control.

Jan 2026 build 2.83.2.1: Migrated from Ignition's legacy historian implementation to the new historian implementation. Upgraded to 8.3 API.

Mar 2026 build 2.83.2.2: Added override fields. Added quarantining back to module, though there is currently no way to get the data back into the forward queue at this time. Changed field descriptions in web form.

June 2026 build 2.83.2.3: Fixed null fields bug introduced by override fields. Queries include prequery and postquery results. Fixed failure to serialize Python arguments bug in scripting functions. Fixed bug that inserted the wrong timestamp for v3 when using the scripting functions.

Known issues:

Data that is quarantined cannot be retrieved through the gateway UI, backend access to the install folder is required.

Historians are registered with the store and forward system, but the controls in the Store and Forward section do not modify any of the historian settings.

When browsing for historical tags against a v3 endpoint, a folder with the name of a tag provider appears, as well as the regular folder with the server name. Clicking into this folder results in generating the same nodes again and again instead of showing the tags.

A future release is expected to address these issues.

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.