MQTT History

MQTT Module History Configuration

Before we get started on the requirements for enabling History on MQTT Engine Tags, one should take note of the two ways that MQTT Engine historical event processing can result in historical inserts into the database:

1. Engine is configured by default, to write historical events directly to the database, via the Historian, bypassing the Tag.
2. Engine can be configured to write historical events to the Tag instead of directly to the Historian. Reasons for writing directly to the Tag are:
    a. To have Tag Events scripts fire when applicable.
    b. If indirectly referencing MQTT Engine tags.
    c. To have alarms triggered when applicable.

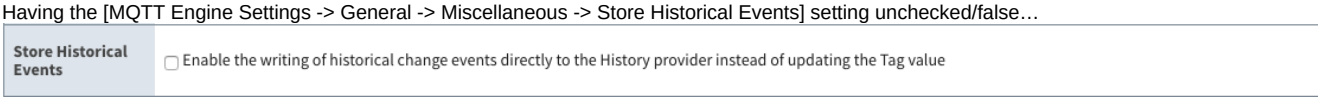
Note: Store and Forward does not guarantee all data is stored and forwarded. There are some edge cases that are not currently handled with regard to data loss in the event of connection failures related to MQTT keep alive timeouts. This window of potential missed data can be reduced by decreasing MQTT Transmission and MQTT Engine configurable keep alive timeouts.

Direct Writes to the MQTT Engine Tag and In-Order History

There are caveats to configuration option 2 above (to write historical events directly to the Tag):

If MQTT Engine is configured to write historical events directly to the Tag, history on the Edge (i.e., Transmission side) must be configured to flush history in order. This means that when the Edge side client comes back online and flushes history, it must flush the oldest historical events first (in order) before sending live Tag changes events to Engine. This is because Ignition will ignore writes to the Tag if the timestamp on the Tag change is older that the current value. Please see the screenshots below for context:

Having the [MQTT Engine Settings -> General -> Miscellaneous -> Store Historical Events] setting unchecked/false…



Requires the [MQTT Transmission Settings -> Transmitters -> [Your Transmitter… Default/Custom] -> Configuration -> In-Order History] setting to be checked/true

MQTT Engine Tag History Configuration

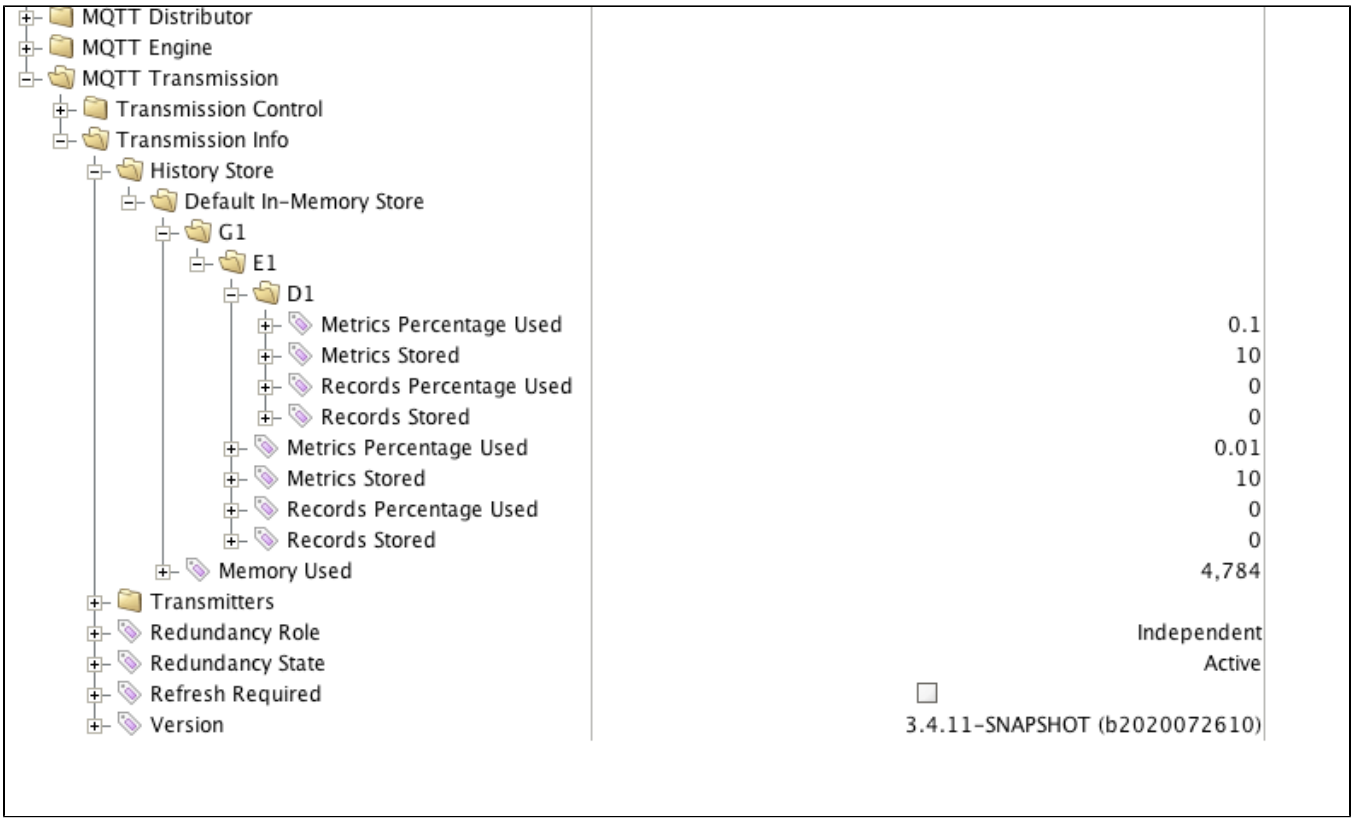
There are some rules for enabling history on MQTT Engine Tags. They are:

1. History must be enabled on the Edge via Transmission side configuration settings.
    a. A History Store must be defined under MQTT Transmission -> History -> History Stores and it must be enabled.
    b. A Transmitter under MQTT Transmission -> Settings -> Transmitters must consume the enabled History Store.
    c. **Ignition 8 Only:** “Enable History Storage by Default” needs to be enabled or the 'CL Store Forward Enabled' custom property must exist on the Edge side Tag in order for historical events to be sent to MQTT Engine.

2. History ***must*** be enabled on the MQTT Engine Tag and the history settings for that Tag must be as follows:
    a. History Enabled: true
    b. Storage Provider: Must be set to an existing Storage Provider
    c. Sample Mode: On Change
    d. Min Time Between Samples: 0 ms
    e. **Ignition 7 Only:** “Timestamp Source” must be set to “Value” (***not*** “System”)
    This is required in order for the historical event to have the timestamp when the value changed on the Edge, instead of when the historical event is being processed by Engine. Having the “Timestamp Source” set to “System” will result in historical events being inserted with the wrong (later) timestamp.
    NOTE: the above history settings will ensure the proper historical event processing by MQTT Engine when writing history directly to the database and when writing history directly to the Tag.

History Stores Metrics & Flushing History

One can determine the current size of the History Store by examining the History Store metric tags under [MQTT Transmission]Transmission Info/History Store. These tags will show the number of historical metrics stored per edgenode/device and how much memory/disk is being consumed by these metrics. These tags update live (count down) as historical data is flushed.



History and Indirect References to MQTT Engine Tags

Configuring history on tags indirectly referencing MQTT Engine tags will work properly if the referencing tag is a **Reference** tag only. Derived, Expression and OPC tags (expose MQTT Engine tag provider through OPCUA server) will not properly store history when MQTT Engine tags are updated with historical data at a high rate of speed. This is a limitation within the Ignition platform and may be addressed in a future release.

In order for historical tag changes at the MQTT Engine tag to propagate to the referencing tag, MQTT Engine must be configured to write historical data directly to its tags and MQTT Transmission must be configured to flush history in-order. See this section above for more details on these configuration requirements.

Additional Resources

- Inductive Automation's Ignition download with free trial
  - https://inductiveautomation.com/downloads/
- Azure Injector download with free trial
  - https://inductiveautomation.com/downloads/third-party-modules
- Questions about this tutorial?
  - Check out the Cirrus Link Forum: https://forum.cirrus-link.com/
  - Contact support: support@cirrus-link.com
- Sales questions
  - Email: sales@cirrus-link.com
  - Phone: +1 (844) 924-7787
- About Cirrus Link
  - https://www.cirrus-link.com/about-us/