

Configuring Tag Historian

Setup Tags to Log

Logging data is easy with Tag Historian Module. Once you have a [database connection](#), just set your Tags to store history. Ignition creates the tables, logs the data, and maintains the database. To learn how to log history for your Tags, go to [Set up Tags to Log](#).



You must first install the **Tag Historian module** so that the standard Tag Historian features are added to Ignition and the Historian Providers show up in the Historian Configuration section of the Gateway.

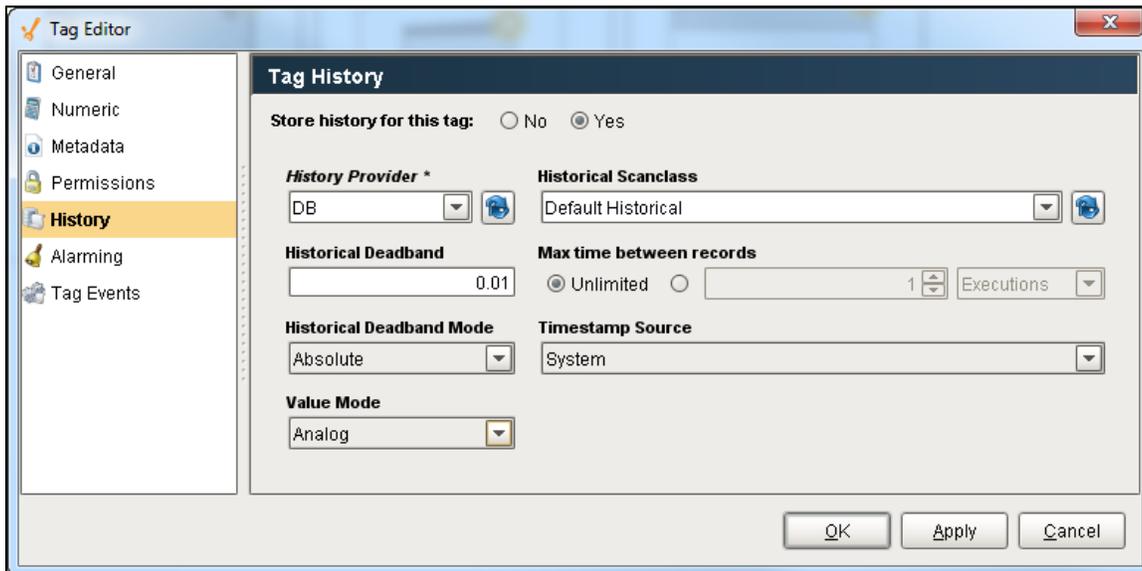
Tag Configuration and Historical Value Generation

The first step to storing historical data is to configure Tags to report values. This is done from the [History](#) page in the **Tag Editor** in the Designer. The properties include a historical [scan class](#), that will be used to check for new values. Once values surpass the specified deadband, they are reported to the history system, which then places them in the proper store and forward engine.

On this page

...

- [Setup Tags to Log](#)
- [Tag Configuration and Historical Value Generation](#)
 - [Historical Scanclass](#)
 - [Value Mode](#)
 - [Max Time Between Records](#)
 - [Timestamp Source](#)
 - [The Deadband, and Analog Compression](#)
- [Add History to Tags in a UDT](#)
- [Historian Providers](#)
 - [Datasource History Provider](#)
 - [Remote History Provider](#)
 - [Tag History Splitter](#)
 - [OPC-HDA Provider](#)
 - [Store and Forward](#)



Historical Scanclass

Historical Scanclass determines how often to record the value on the Tag. It uses the same [Scan Classes](#) that dictate how often your Tags should execute. Typically, the Historical Scanclass should execute at the same rate as the Tag's Scan Class or slower: if a Tag's Scan Class is set to update at a 1,000ms rate, but the Historical Scanclass is set to a scan class that runs at 500ms rate, then the Tag History system will be checking the Tag's value twice between normal value changes, which is unnecessary.

Evaluate on Change

Additionally, the Historical Scanclass may be set to the **<Evaluate On Change>** option, which records every time the Tag changes value. This is useful if you wish every new value Ignition sees on the Tag.

Note: This scanclass setting will ignore the Max Time Between Records setting.

Value Mode

The value mode, **analog** or **discrete**, dictates the type of value that the Tag represents, and will affect how the deadband is applied to values, and how interpolation should be performed when querying. Interpolation is the method in which the Tag Historian query system generates values for a Tag when the desired time does not fall directly on a sample timestamp.

Discrete

- **Storage** - The deadband will be applied directly to the value. That is, a new value (V_1) will only be stored when: $|V_1 - V_0| \geq \text{Deadband}$.
- **Interpolation** - The value will not be interpolated. The value returned will be the previous known value, up until the point at which the next value was recorded.

Analog

- **Storage** - The deadband is used to form a corridor along the trajectory of the value. A new value is only stored when it falls outside the previous corridor. When this occurs, the trajectory is recalculated, and a new corridor formed. See below for an example.
- **Interpolation** - The value will be interpolated linearly between the last value and the next value. For example, if the value at Time_0 was 1, and the value at Time_2 is 3, selecting Time_1 will return 2.

Max Time Between Records

Normally Tag Historian only stores records when values change. By default, an "unlimited" amount of time can pass between records – if the value doesn't change, a new row is never inserted in the database. By modifying this setting, it is possible to specify the maximum number of scan class execution cycles that can occur before a value is recorded. Setting the value to 1, for example, would cause the Tag value to be inserted each execution, even if it has not changed. Given the amount of extra data in the database that this would lead to, it's important to only change this property when necessary.

Timestamp Source

When a Tag executes, there are two possible timestamps that can be observed: the time associated with the data, and the time that the Tag was evaluated. The first case is generally only interesting when the value is provided by an OPC server. In most cases, the time provided by OPC, which in Ignition is referred to as the "Value" time, will be very close to the system time. Some servers, however, either due to their location or how they function (history playback, for example), will provide times that are very different than the current time.

It is generally desirable to store the System time, as it is the time that the value was actually observed by the system, and it creates a uniform timeframe for all realtime data. However, in the later case, it is necessary to store the time provided by the OPC server. Using the Value timestamp source has several consequences: the system is no longer able to validate the Tag quality against the scan class' execution, and Tag value interpolation will behave differently.

The validation of the scan class execution is generally not a concern when recording historical playback data. Interpolation only occurs when the value mode is Analog, and when there is not a value for every time window. Using System time, the value is only interpolated during the last "scan class execution window", that is, one scan class timeframe before the next value. Using Value time, however, the value is interpolated for the entire time between two data points.

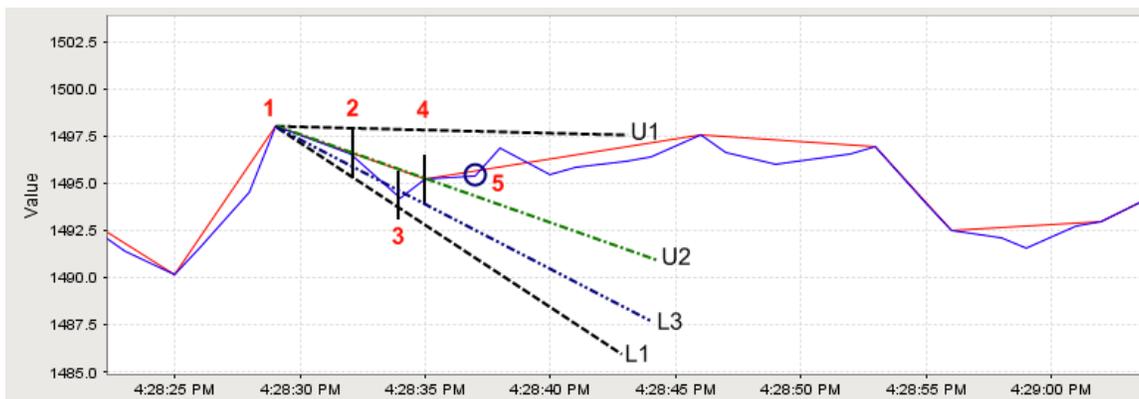
The Deadband, and Analog Compression

As described above, the deadband value is used differently depending on whether the Tag is configured as a Discrete Tag, or an Analog Tag. Its use with discrete values is straight forward, registered a change any time the value moves +/- the specified amount from the last stored value. With Analog Tags, however, the deadband value is used more as a compression threshold, in an algorithm similar to that employed in other Historian packages. It is a modified version of the 'Sliding Window' algorithm. Its behavior may not be immediately clear, so the following images show the process in action, comparing a raw value trend to a "compressed" trend.



In this image, an analog value has been stored. The graph has been zoomed in to show detail; the value changes often and ranges over time +/- 10 points from around 1490.0. The compressed value was stored using a deadband value of 1.0, which is only about .06% of the raw value, or about 5% of the effective range. The raw value was stored using the Analog Tag mode, but with a deadband of 0.0. While not exactly pertinent to the explanation of the algorithm, it is worth noting that the data size of the compressed value, in this instance, was 54% less than that of the raw value.

By looking at one specific sequence, we can see how the algorithm works:

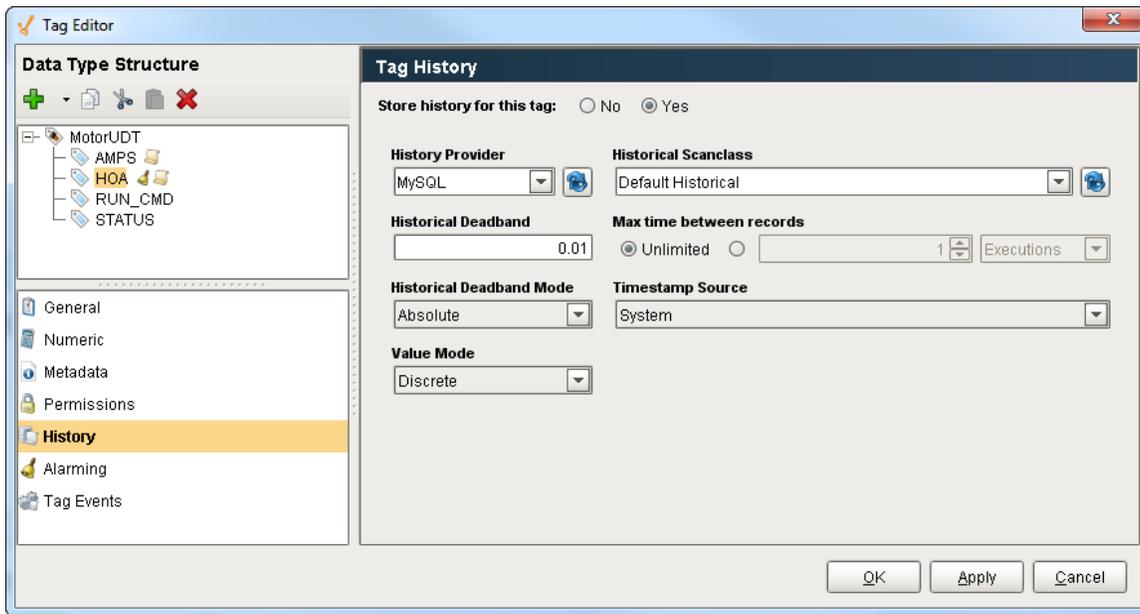


The sequence starts with the second stored compressed value on the chart.

1. A value is stored. No further action is taken.
2. The next value arrives. A line is made through the value, with the size of the specified deadband value. A line is projected from the last stored value to the upper (line U1), and lower (line L1), bounds of this new value line. This establishes the initial corridor.
3. A new value arrives. The same procedure is taken, and new lines are created. However, only lines that are more restrictive than the previous are used. In this case, that means only line U2, the new upper line.
4. Another value arrives, causing a new lower line (L3) to be used.
5. Finally, a value arrives that falls outside of our corridor. The last received value (value 4) is stored, and the process is started again from that point.

Add History to Tags in a UDT

Setting up UDTs to log history data works on the same principle as [creating UDTs](#). You dramatically reduce the amount of work when setting up UDTs to log history data so that all instances of that UDT will log data without having to edit each individual instance. To learn how to set the UDT to log history, refer to [Add History to Tags in UDT](#).



Historian Providers

Tag History Providers are what drive the historian in Ignition. They typically require some sort of database that stores the data to be recalled at a later time in a trend or report. They are setup or configured in the Configure tab off the Gateway Webpage by going to **Tags > History**.

Historical Tag Providers

Provider Name	Enabled	Type	Description	Status	
NewConnection	true	Datasource History Provider		Running	edit

[→ Create new Historical Tag Provider...](#)

Datasource History Provider

The default history provider that automatically gets created for each new database connection that gets established and will be removed if that database connection is deleted. While it can be configured to allow for some different storage options, it is setup with a 1 month partition by default.

Remote History Provider

This provider provides a link to a provider on another Gateway. To use this type of provider, there must be an established Gateway Network. It can be used just to read history values that the remote provider has logged, or it can be configured to allow the local Gateway to write values to the remote Gateway.



INDUCTIVE
UNIVERSITY

Remote History
Provider

[Watch the Video](#)

Tag History Splitter

This provider combines two separate providers into a single new provider. When setting up a Tag to store history, selecting this provider will write the same data to both providers that it has selected. The Tag History Splitter is useful for automatically creating a backup of your data, or for reading history from two separate providers. Learn more about setting up the [Tag History Splitter](#) here.

OPC-HDA Provider

Establishes a connection to an [OPC-HDA Server](#) to read history data that may be stored there from a third party. Ignition can't write to this type of history provider.



This requires the OPC COM module be installed.

Store and Forward

The Store and Forward system provides a reliable way for Ignition to store historical data to the database. The Store and Forward system is not exclusively part of Tag History, but systems such as the Tag Historian and Transaction Groups use this to prevent data loss and store data efficiently using a record cache.

It is important to configure your Store and Forward system when using Tag history. For more information, see the following sections:

- [Store and Forward](#)
- [Using Store and Forward](#)
- [Configuring Store and Forward](#)
- [Controlling Quarantine Data](#)



Related Topics ...

- [Tag Historian and Tables](#)
- [Understanding Transaction Groups](#)

In This Section ...