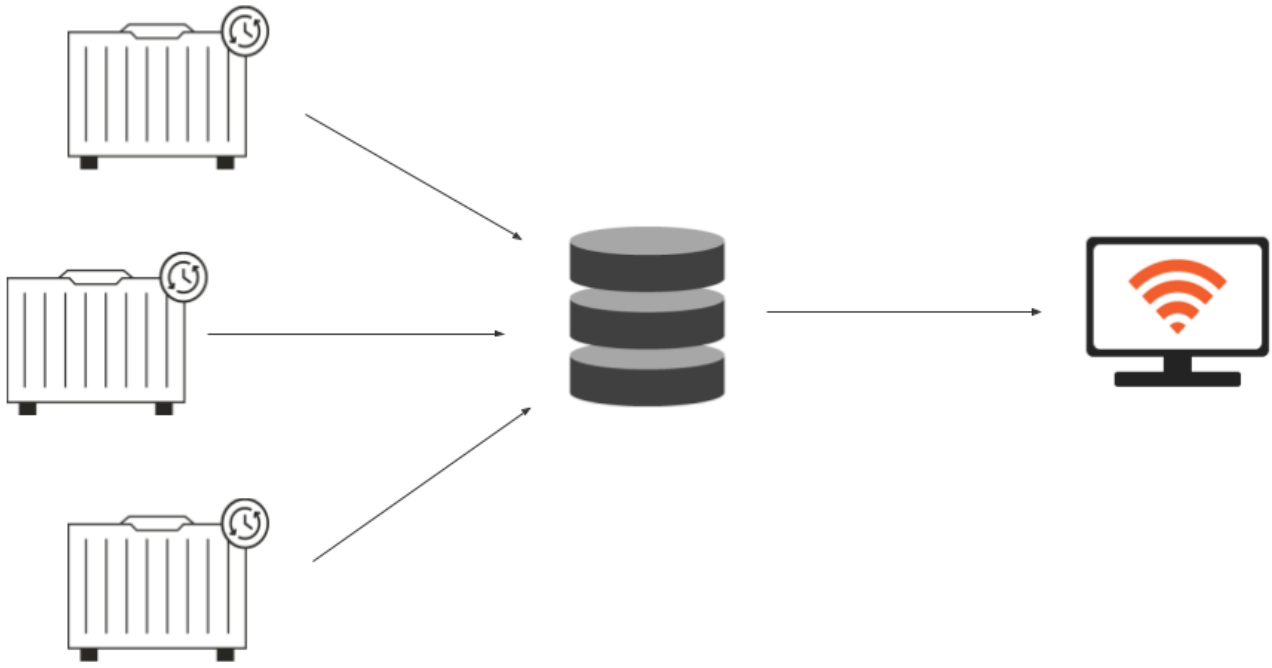
www.dewesoft.com - Copyright © 2000 - 2024 Dewesoft d.o.o., all rights reserved.

Dewesoft Historian Installation and Usage



Dewesoft Historian

Dewesoft Historian is a DewesoftX plugin solution that enables continuous measurement and storing into a database from Dewesoft Measurement Units as well as the access to the database using a web browser or DewesoftX software. Dewesoft Historian consists of several software components that usually run on different systems (measurement units, servers, client PCs).

Measurement Units, databases and clients within the Dewesoft Historian system communicate over the MQTT protocol which enables each component to run on a separate computing system at any location with internet access.

To begin this pro training, you will need to download the DewesoftX software and activate an evaluation license.

Essential Requirements:

- **DewesoftX software**
  - **Download the DewesoftX software here.**
- **DewesoftX evaluation license**
  - Request the license here.

Resources

- **Docker Desktop**
  - Download Docker Desktop here.
- **Dewesoft Historian Docker Package**
  - Download Historian Docker Package here.
- **Dewesoft Historian Manual**
  - Download Dewesoft Historian Manual pdf here.

MQTT Protocol

Before we begin this training, it's essential to have at least a basic understanding of how the MQTT protocol works. This knowledge will help you better comprehend the operation of the historian system.

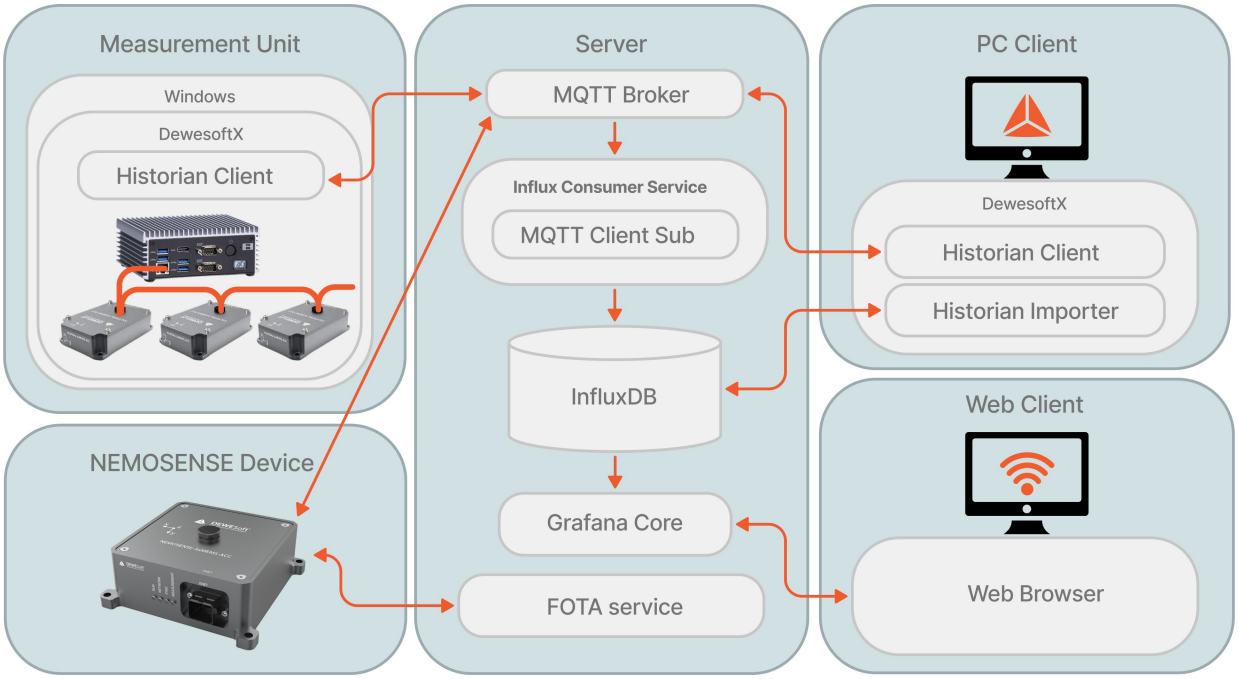
To gain a better understanding of the system, we recommend reading about MQTT basics or watching a short video on the MQTT protocol.

In the MQTT protocol, communication is established between two main entities: the **publisher** and the **subscriber**, both connected through a **broker**.

- **The broker** routes messages from the publisher to subscribers.
- **Publisher** sends messages to the broker
- **Subscriber** listens for messages from the broker

The Historian Client plugin can function as both a publisher and a subscriber on the same system. However, in a typical use case, one instance is used solely as a publisher, while another instance is used solely as a subscriber.

System Overview



- **Historian Client** is a DewesoftX plugin running on a measurement unit. It is used to publish or subscribe the data, using the MQTT protocol, to/from the MQTT broker.

- **MQTT broker** is the central component that temporarily holds the data published by Historian Clients running on the Measurement units and serves the data either to the MQTT client subscribers that can be either Influx consumer service or Historian Clients (DewesoftX instances).

- **Influx consumer service** subscribes to the MQTT broker and sends the data to the InfluxDB database, where the data is permanently stored.

- **InfluxDB time series database** permanently stores the data. InfluxDB is a leading time series database solution designed with measurement applications in mind. Its open source version is used within the Dewesoft Historian solution.

- **DewesoftX Historian PC Client** is an instance of Dewesoft software that runs on client PCs and includes two plugins: Historian Client and Historian Importer. Historian Client plugin operates in Dewesoftâ€™s Measure mode and has the role of displaying live data from the measurement units.

- **Historian Importer plugin** operates in Dewesoftâ€™s Analysis mode and allows the user to load historical data from the database into a Dewesoft .dxd data file for analysis.

- **Grafana GUI** is a visualization platform for measurement data that runs in a web browser. Its open source version is used within the Dewesoft Historian software solution. In its simplest form it only requires the Grafana Core installation running on the server which allows users from any web browser to connect to and visualize the data from the database.

However, Dewesoft Historian license package includes the following enhancement of the solution:

- **Dewesoft Grafana plugins** which provide user interfaces not available in a standard Grafana such as 2D graph (called Vector) to display FFTs and similar data channels.

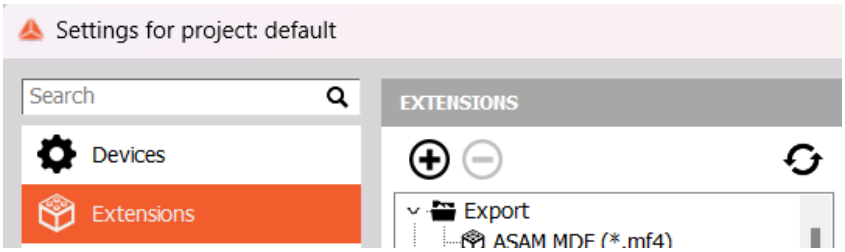
We will access a publicly available Dewesoft Historian instace at **historian.dewesoft.com**, port 1883 (MQTT) or 8883 (secure MQTTS) later in the training.

Installation

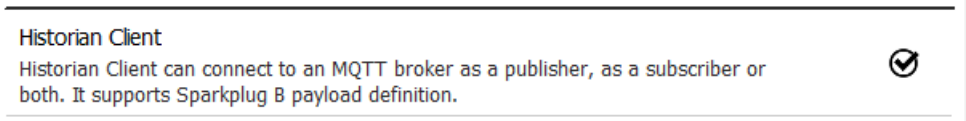
To stream the data from DewesoftX software, the DewesoftX Historian Client plugin is required. As of version 2023.3 onwards, the DewesoftX installer already includes the aforementioned plugin. **In older DewesoftX versions the plugin is NOT supported!**

To enable the Dewesoft Historian Client, follow these steps:

- **Open DewesoftX** and navigate to **Options** (upper right corner).
- Go to **Settings** and select **Extensions**.
- Search for **â€œHistorian Clientâ€** in the list of extensions. If the plugin is not listed, click the **+ icon** at the top.



- Change the filter to **â€œAllâ€** and search for **Historian Client** in the list. Enable it, and the plugin should now be activated.

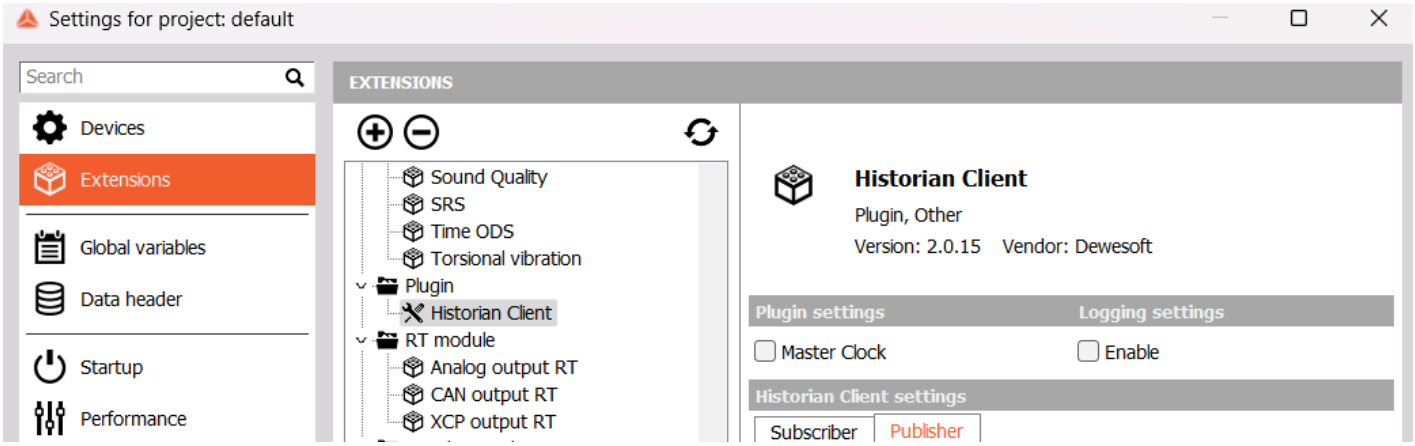


If the plugin is still not found in either list, you need to update the DewesoftX plugins:

1. Go to **Options** > **Add or remove plugins** to manually update the plugins.
2. Alternatively, run the **DewesoftX installation .exe** again.

Configuration

After successful installation Dewesoft Historian Client should be listed in extensions list. Select it to open the Historian Client settings.



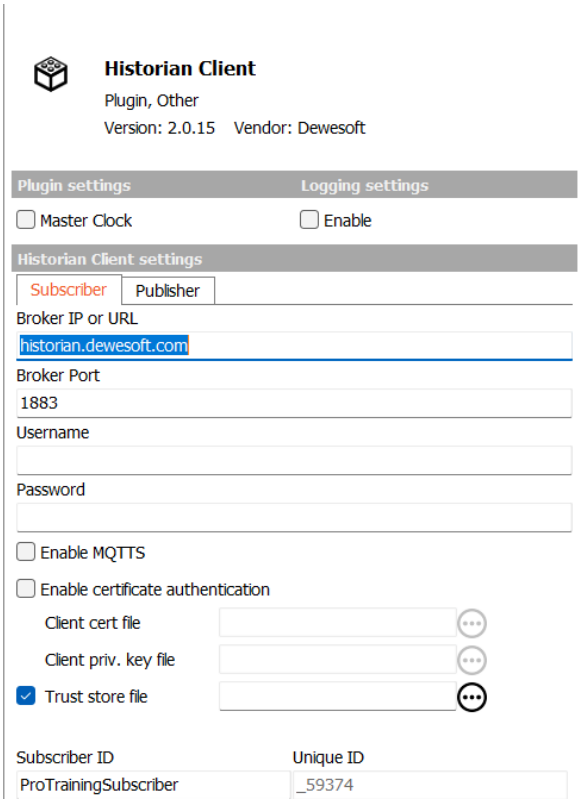
First, we need to configure both the subscriber and publisher with the correct broker URL/IP and port. Setting the subscriber to the right IP address defines where the data is being received from. On the other hand, configuring the publisher defines where to send (or publish) the data, ensuring the measurements or messages are properly transmitted to the selected broker for distribution to subscribers. This configuration ensures clear communication paths in the data flow process.

For the purpose of this training, we'll be using a broker on the publicly available Dewesoft Historian server: **historian.dewesoft.com**.

Subscriber Setup

The **subscriber** listens for messages from the broker on specific topics. To set it up, follow the instructions below:

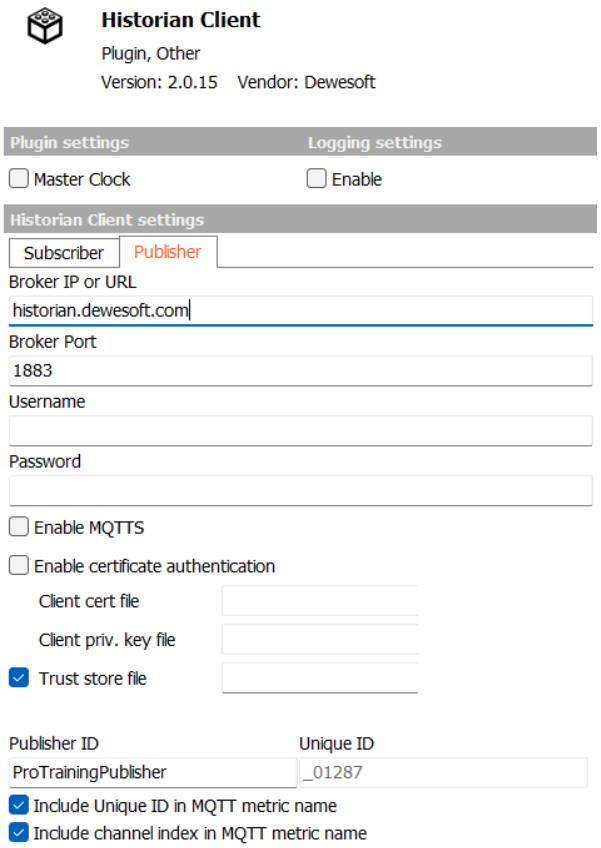
- set **Broker URL/IP** to **historian.dewesoft.com**
- set **Port** to **1883**
- set **Subscriber ID** to **ProTrainingSubscriber**
- Optionally username and password if required by the broker. (not required on historian.dewesoft.com)



Publisher Setup

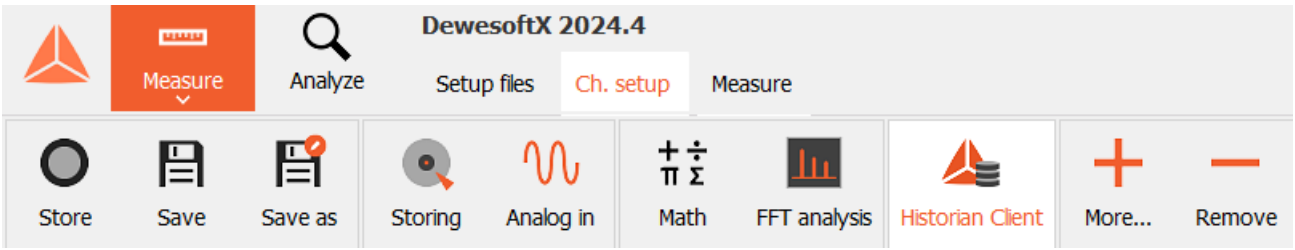
The **publisher** sends messages to the broker on specific topics, which are delivered to subscribers. To set it up, follow the instructions below:

- set **Broker URL/IP** to **historian.dewesoft.com**
- set **Port** to **1883**
- set **Subscriber ID** to **ProTrainingPublisher**
- Optionally username and password if required by the broker. (not required on historian.dewesoft.com)



To confirm the configuration and verify a successful connection to the server, check the logs by clicking â€œInformationâ€ in the top right corner or by going to **Options > Tools > Event Viewer**, ensuring both â€œShow Warningsâ€ and â€œShow Informationâ€ are enabled.

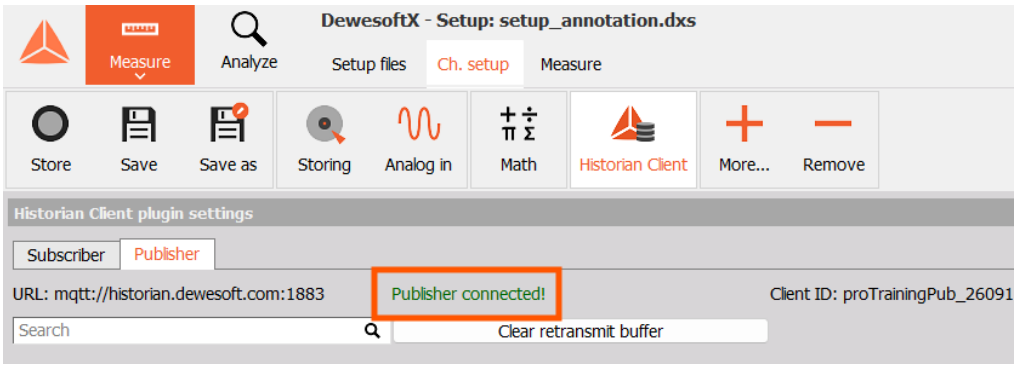
Once the Historian Client plugin is enabled and configured, set your DewesoftX to measure mode and select channel setup. The Historian Client icon should appear.



Now, we can observe the simplicity and versatility of the Historian plugin. It is divided into two main components:

- **Subscriber** section displays a list of all channels published to the Historian MQTT broker. Here, users can easily subscribe (select) to any offered data channel being transmitted by different publishers.
- **Publisher** window allows users to create publishing channels within the same plugin for data transmission.

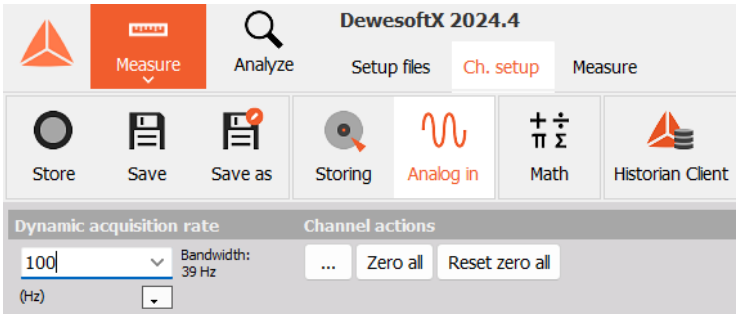
Inspect the Subscriber and Publisher sections of the Historian client, and check their status sections to verify that both are successfully connected.


Measure
Analyze
DewesoftX - Setup: setup_annotation.dxs
Setup files
Ch. setup
Measure
Store
Save
Save as
Storing
Analog in
Math
Historian Client
More...
Remove
Historian Client plugin settings
Subscriber
Publisher
URL: mqtt://historian.dewesoft.com:1883
Publisher connected!
Client ID: proTrainingPub_26091
Search
Clear retransmit buffer

Create a Publisher Channel

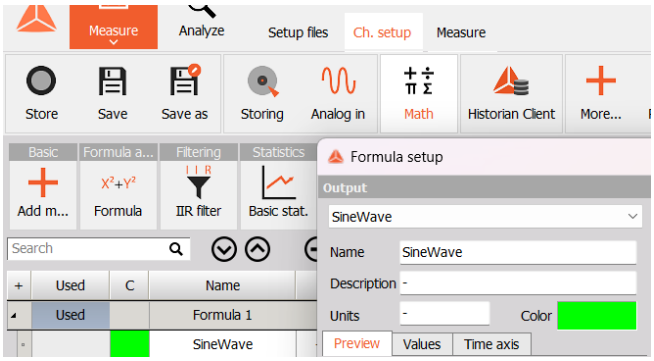
Now, letâ€™s create a simulated publisher channel!

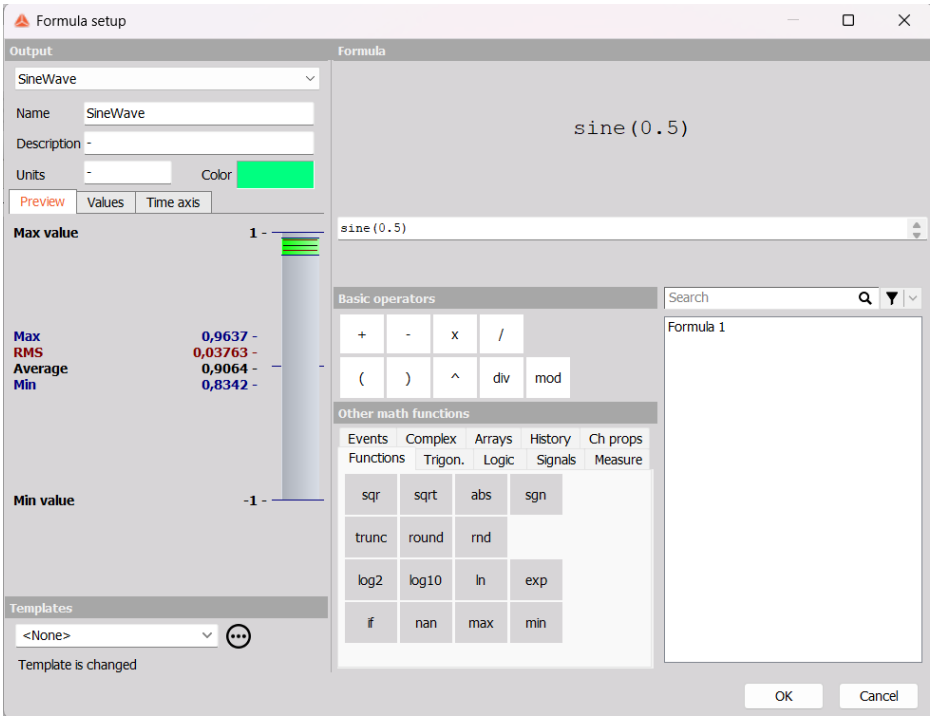
First, open the "Analog In" section in the channel setup and set the sample rate to 100 Hz.



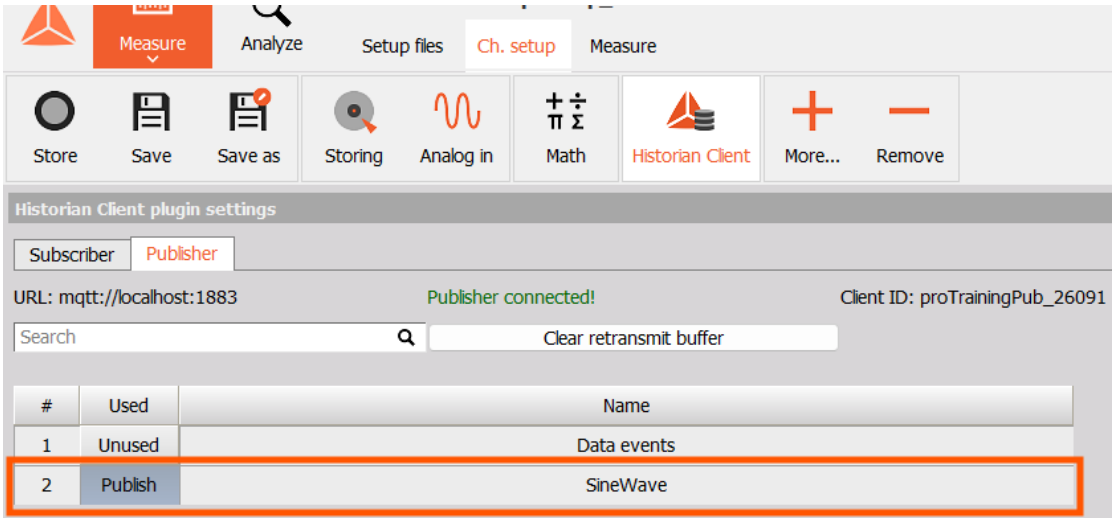
To create a simulated measurement, open the "Math" section and add a sine formula channel to simulate a sine wave. Once you've completed this step, confirm the creation of your simulated measurement channel.

Name the formula in a way that will make it easier and faster for you to find later.



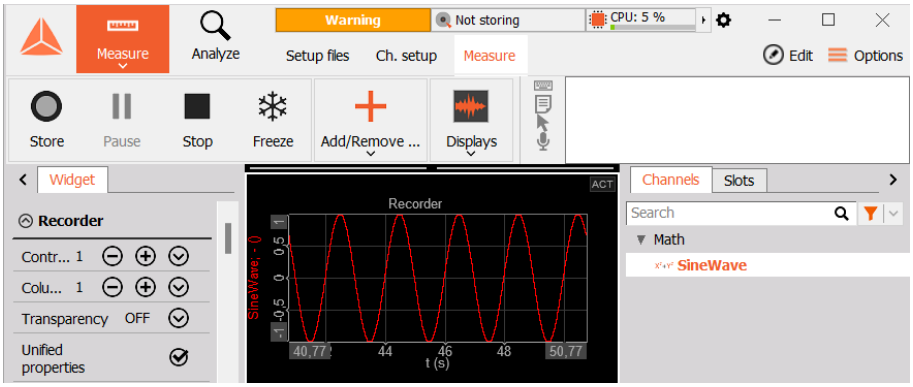


Next, switch back to the Publisher section in the Historian Client plugin. Your newly created simulated measurement channel should now be visible.



Letâ€™s go live!

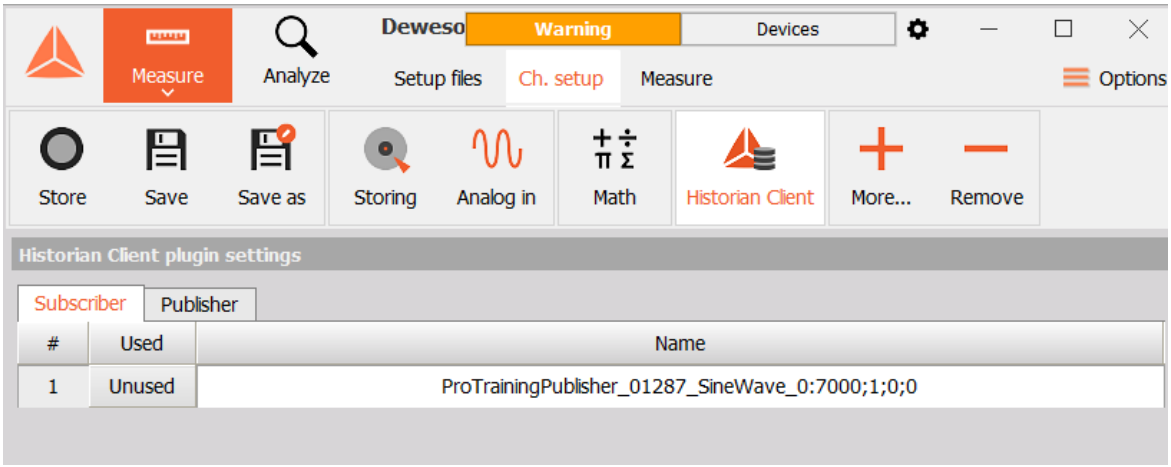
The final step is to start publishing data from your measurement channel. Locate your simulated measurement channel in the list of publishers and enable it by clicking the button in the "Used" column. Once the button changes to "Publish," go to the "Measure" window, where you should see your simulated sine wave.



You are now live, and your channel should now be sending data to the broker on the server.

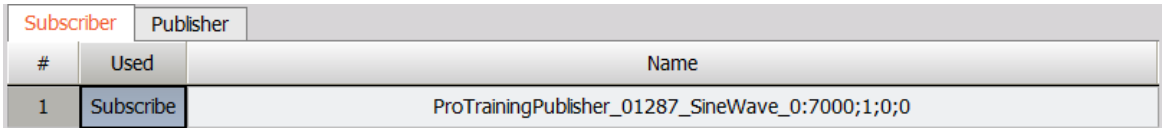
Monitor Your Published Channel

Your published simulated channel is now up and running. The next step is to subscribe to it (i.e., read the data). Since both the Subscriber and Publisher are configured to the same broker URL/IP, your simulated channel should now be visible in the Subscriber tab of the Historian Client.



If you cannot find your simulated channel, ensure that both the **Subscriber** and **Publisher** are configured to **the same Broker address and port** (refer to the Configuration section of the training). Additionally, verify that **publishing is enabled in the Publisher tab** by checking if the channel is set to "Publish".

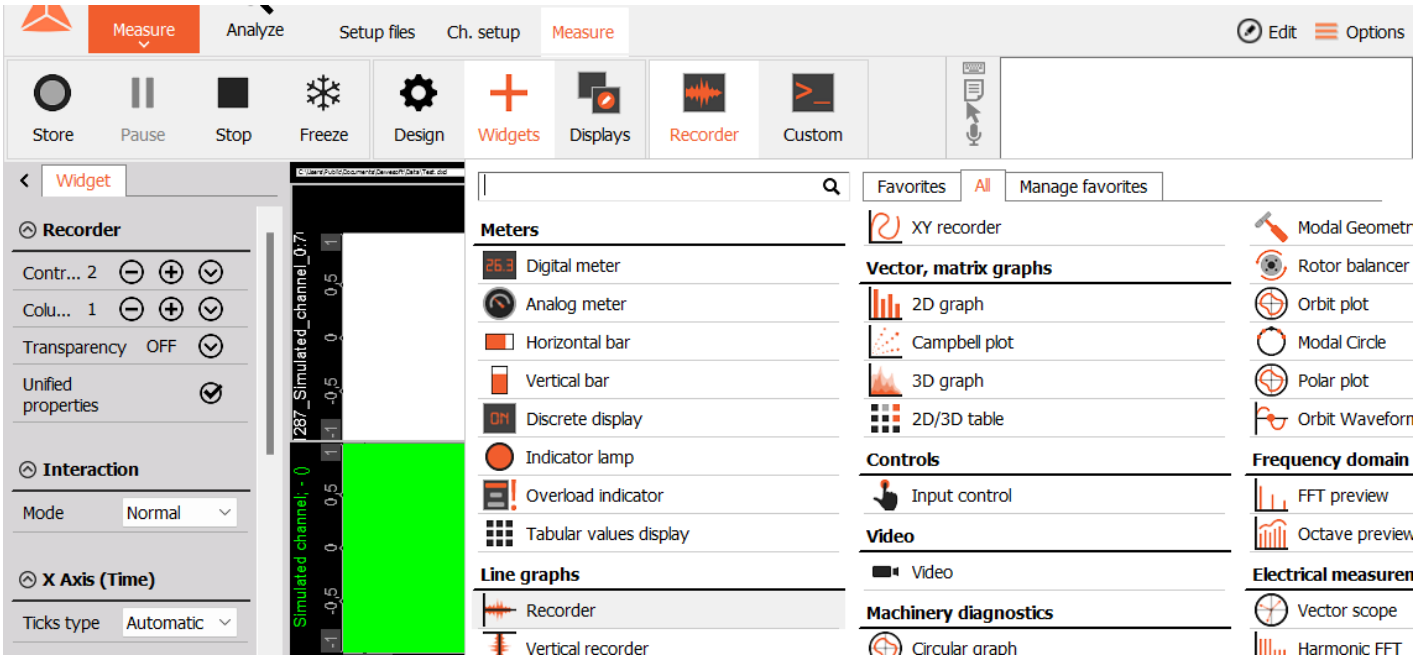
To subscribe to your simulated channel, locate it in the channels list and enable it by clicking the button in the "Used" column. Once the button switches to "Subscribe," you will begin fetching data from the publisher through the broker in real time.



Congratulations! You are now simulating a measurement and functioning both as a publisher and a subscriber. On the **publisher** side, **you are publishing your simulated channel** to historian.dewesoft.com, while at the same time, acting as a **subscriber** by connecting to the same server and **fetching the data** youâ€™ve just published.
Although the published measurements are being stored in a database on the remote server, the data flow in publish/subscribe process bypasses the database entirely. Instead, data is transmitted directly from the **publisher**, through the **broker**, to the **subscriber**.

Visualize Subscribed Channel in DewesoftX

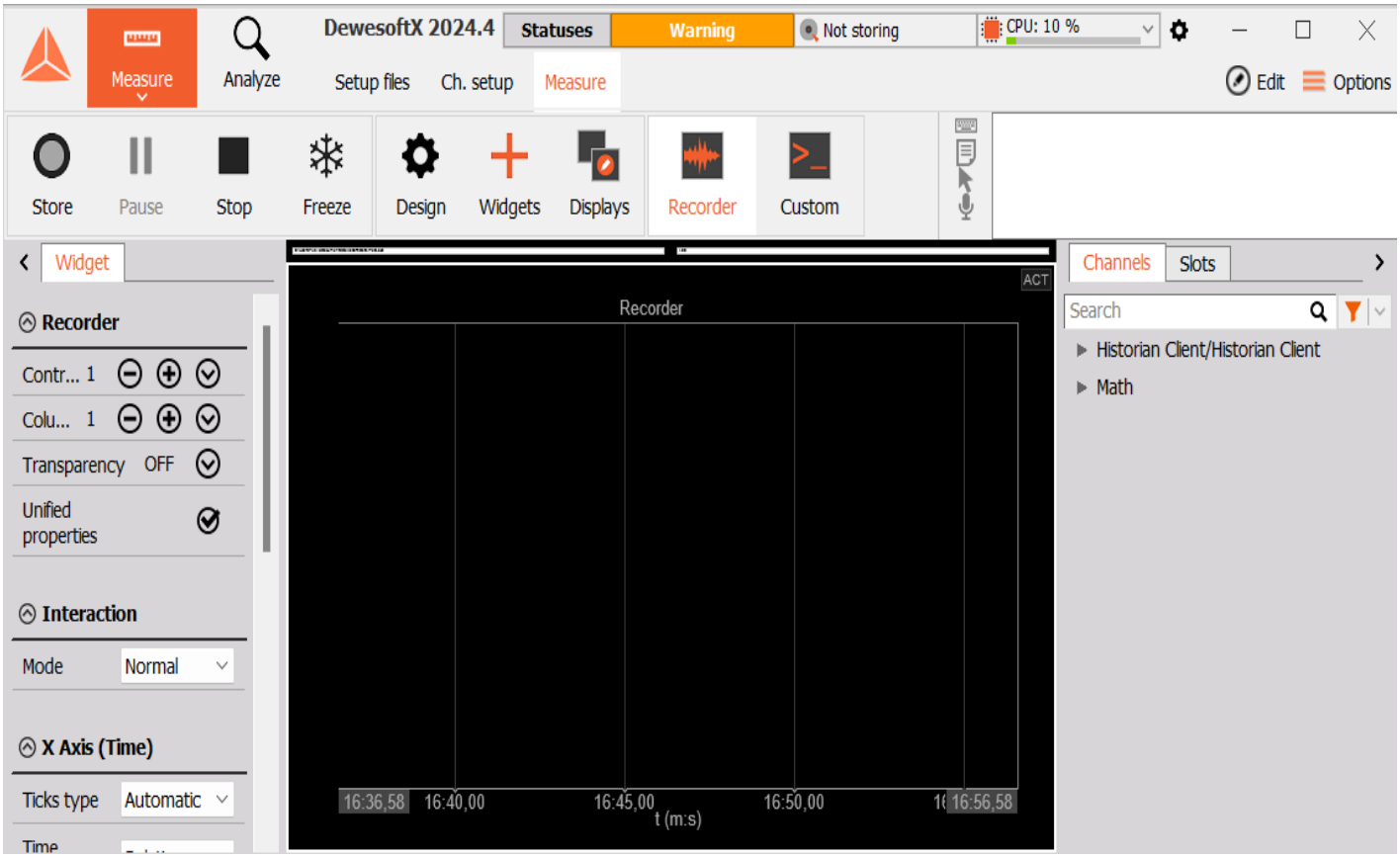
Once you're subscribed to your simulated channel, the fetched data needs to be visualized. With the channel active (subscribed), open the **Measurement section**. To visualize live data, we will use the Recorder tool. By selecting "**Recorder**," graphical visualizations for all active channels should appear automatically. If this does not occur, you can manually add a recorder by opening the Widgets menu and selecting "Recorder."



Recorder

Once the recorder window is open, it can be divided into three vertical sections:

- **Recorder Settings** (leftmost section)
- **Graph Section**
- **Channel Selection Section**



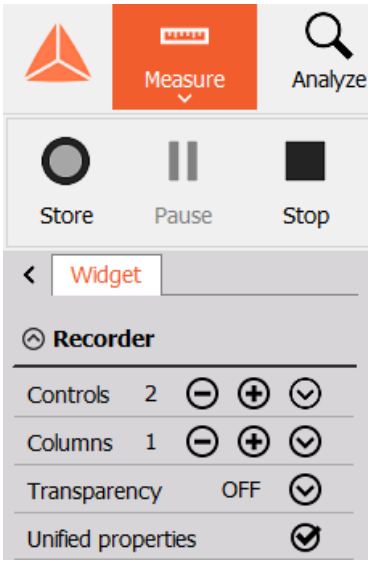
Make sure you are subscribed to the channel you have published. You should now have two active instances of the same channel:

- The **first instance is your local simulated sine wave,** which runs directly in DewesoftX.
- The **second instance is the same sine wave, but now the data flows through MQTT**. This means that the sine wave data is published from DewesoftX to the MQTT broker, and then subscribed back into DewesoftX.

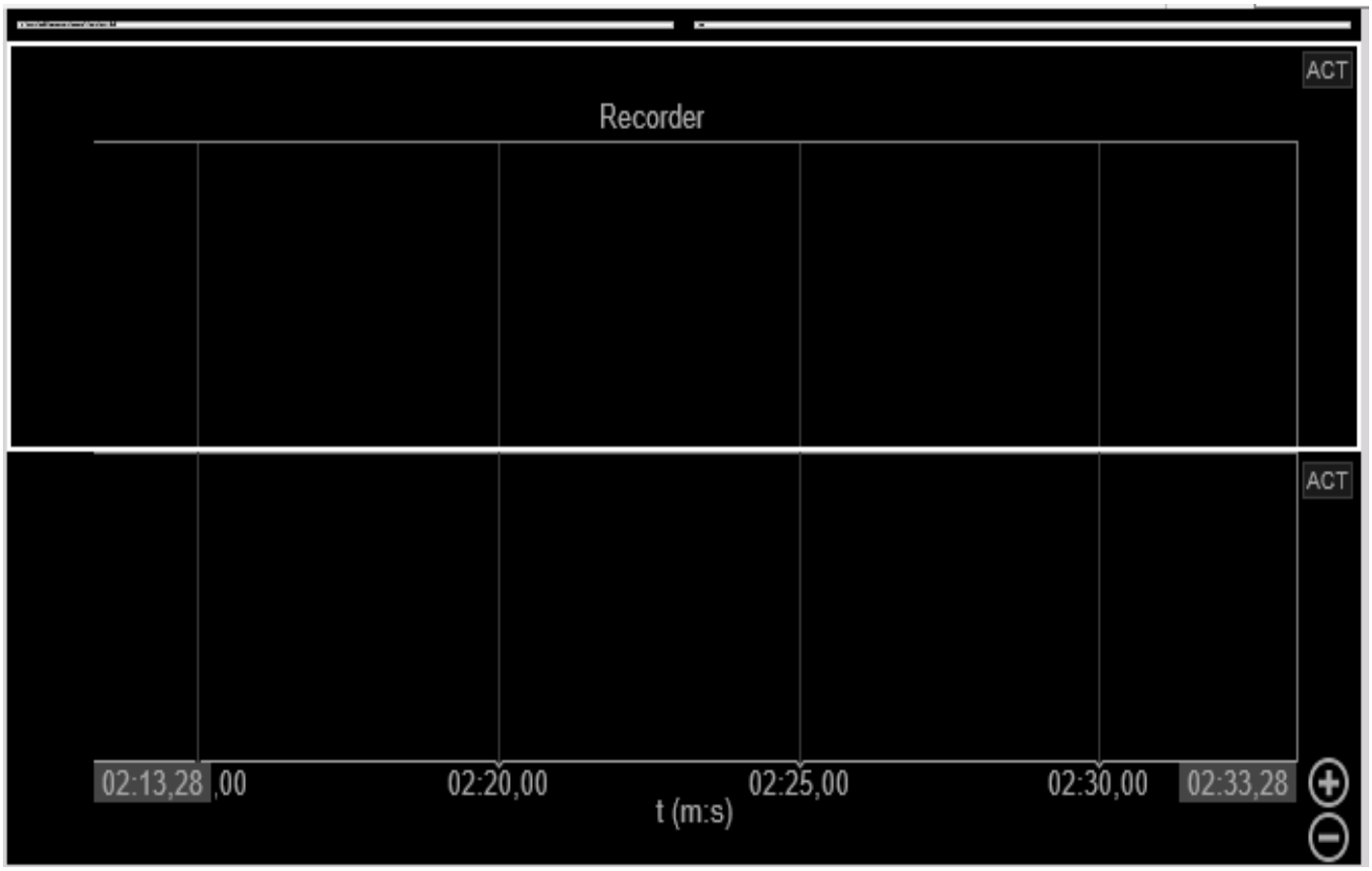
To display both the subscriber and local instance of the channel, go to the **Channel Selection Section** and switch to the channel tab. You should see your channels divided into two groups:

- **Math Group:** This contains all your local math channels, including your local simulated sine wave channel.
- **Historian Client:** This contains Historian Client channels.

Now, to compare the local channel with your published channel, youâ€™ll need two recorders. If you donâ€™t have two on the screen, use the Controls section to add or remove recorders by selecting the "+" or "-" controls option.



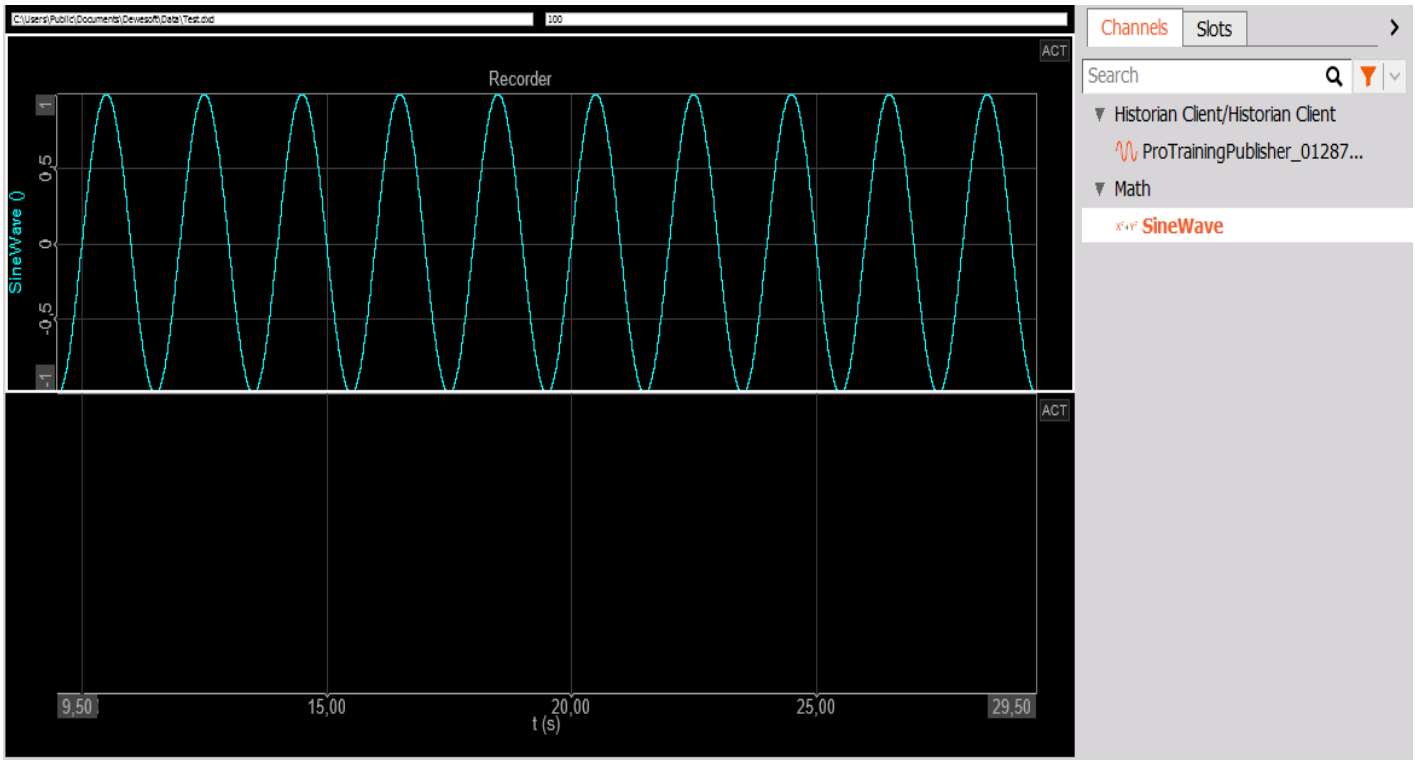
Your display should look something like this:



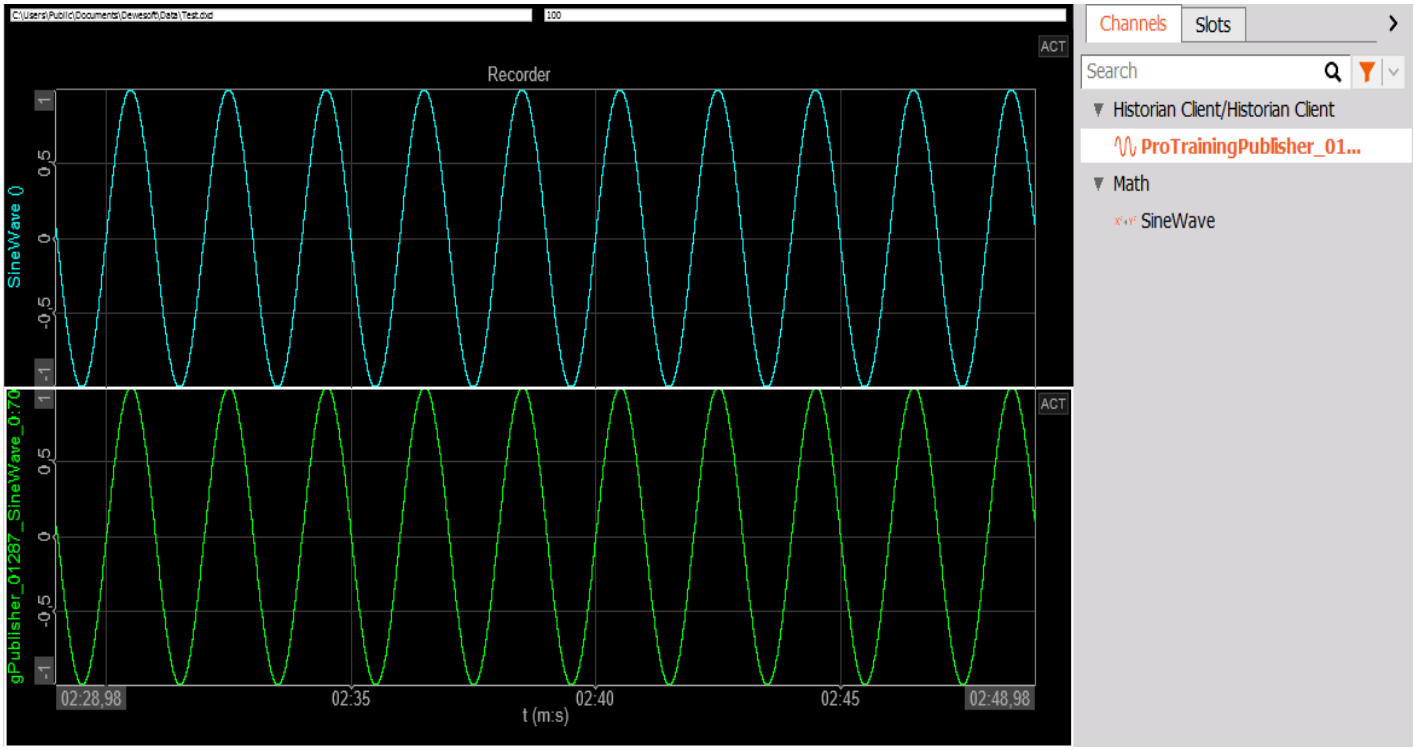
As discussed earlier, both your local and subscriber channels are located in the Channel Selection Section. To add a channel to a recorder, follow these steps:

- **Select the Recorder**: Click directly on the recorder you wish to use. You will know it is selected when a wider white border appears around it.
- With the recorder selected navigate to the Channel Selection Section. In the Math section of the Channel Selection, **find your simulated sine wave channel and select it** (When the background of the channel name turns white, it means the channel is selected.)

At this point, your selected recorder should display your local simulated sine wave channel and should look something like this:



To add your subscribed channel to the second recorder, follow the same steps. The end result should look like this:



If we now examine the end result, we can visually observe our local instance of simulated sine wave channel running alongside the same instance, which is being published by our publisher to the broker and then fetched by our subscriber back into DewesoftX.

This process highlights the flexibility of using a single platform to manage both publishing and subscribing, enabling you to

send, retrieve, or handle both data types within one unified system. However, itâ€™s important to note that in a typical use case, subscription is usually done on a different system, as this configuration is more practical. Performing both actions on the same system is generally intended for training purposes.

Note: Both channels can be added to the same recorder, as it can handle multiple channels. However, for the purpose of this pro training, we'll separate our measurements into two recorders for better visualization. This allows for simultaneous comparison of different data streams on the same graph.

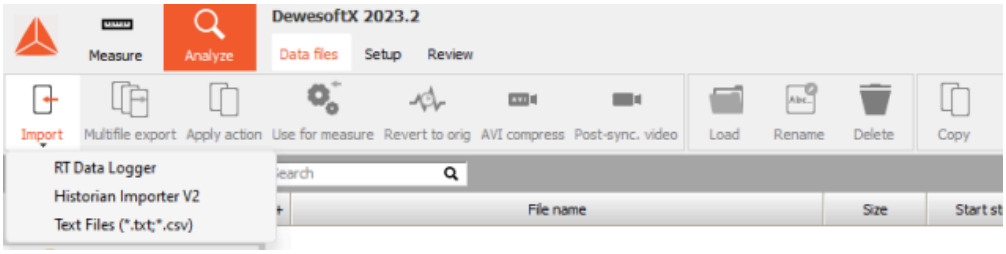
DewesoftX Historian Importer

For more advanced analysis, DewesoftX's Analyze mode provides an extensive set of tools. Since our measurements are stored in a database, the Historian Importer enables us to retrieve this data and convert it into the native .dxd format. Once the data is imported locally into DewesoftX, we can leverage all of its advanced analysis features for thorough and comprehensive examination.

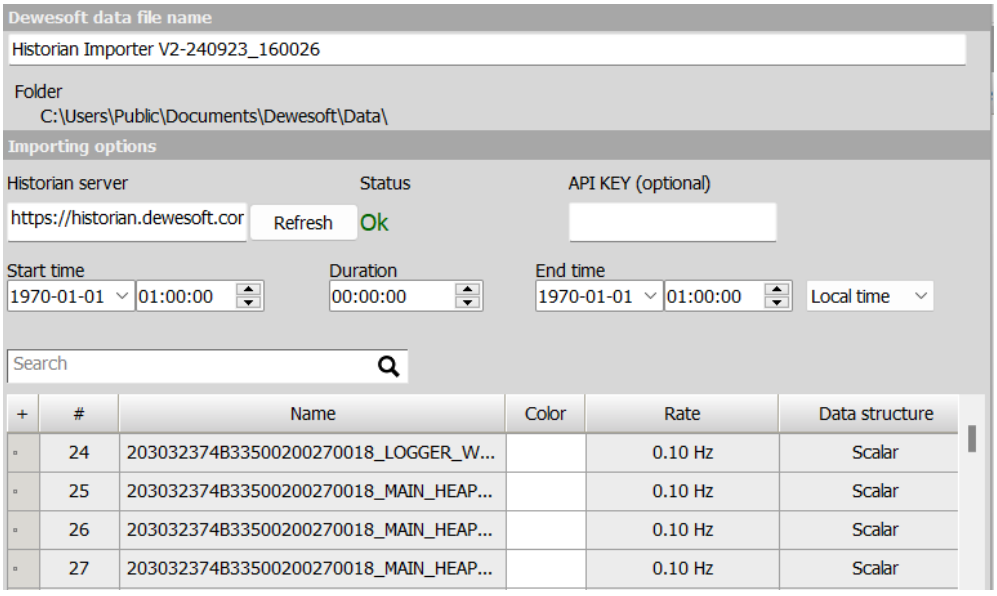
To access the database, you need to know the URL or IP location of the database. To try the Dewesoft Historian importer, you can use **https://historian.dewesoft.com:8082**.

If you plan to fetch data from your local server (refer to the Server section of this training), ensure that you have already published some measurements that were saved into the database. Otherwise, the Historian Importer wonâ€™t be able to find any data to fetch, as there would be none stored yet.

To import data using the Historian Importer, DewesoftX needs to be in Analyze Mode. Click on the Import icon to open the drop-down menu. Select Historian Importer.



After selecting the Historian Importer, the Historian Importer window will appear. In this window, you will find various importing options to configure and manage your data import process.



The most crucial configurations involve selecting the appropriate database address and ensuring a successful connection.

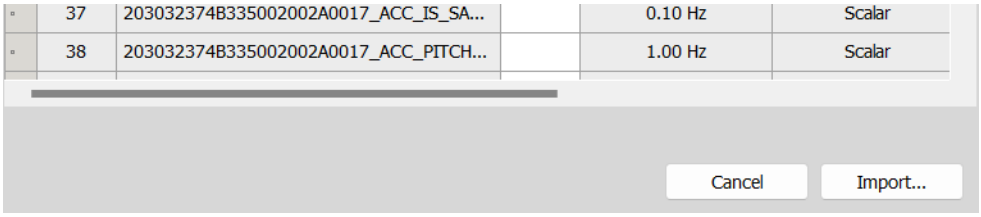
- **Public Historian Server**: Set the server to "**https://historian.dewesoft.com:8082**" to connect to the public Historian server database.
- **Local Historian Server** (Docker): If you're running Docker locally, set the server to " **https://localhost:8082**" to connect to your local Historian server database.

Once the address is set, additional configurations include:

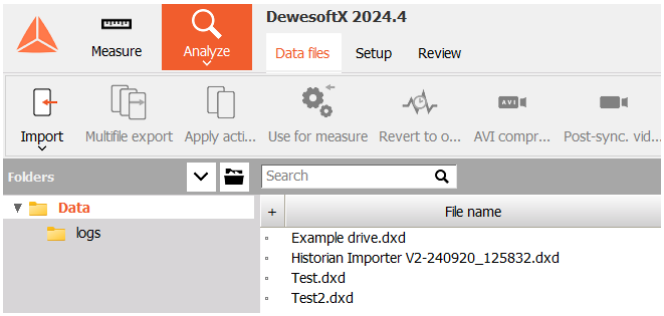
- **Status**: After entering the Historian server IP and clicking Connect, green "OK" letters will appear if the connection to the database is successful.
- **Start time**: Select the start time and date of the data you want to import.
- **End time**: Select the end time and date of the data you want to import.
- **API key**: You may also set an API key for authentication if required by your server. Ensure that you have the correct API key to access the database.

After configuring the Importing options select the channels you'd like to import from the database. This can be found in the table below the Importing options. The table contains the name, rate, data structure and data type of all the channels stored in the database.

After selecting the channel you would like to import, the Start time will automatically update to the time when the acquisition of that channel began. This helps ensure you're retrieving data from the correct time period for the selected channel.



After selecting the channels you'd like to import, click on the Import button. A new data file containing the data from the selected channels will be created. After importing, **Analyze Mode** should automatically open. If this doesn't happen, you can manually view and open your imported files by navigating to the Data Files section, where all your .dxd files are stored, and selecting the imported file. By default, Historian imported files are saved under the name "Historian Importer."



Congratulations! Youâ€™ve successfully used the Historian Importer to fetch the selected channel data from the database. While data analysis is beyond the scope of this training, we strongly recommend checking out our Data Analysis Training for a deeper dive into analyzing your imported data.

Setup Your Own Server

Up to this point, we have been using a public Historian server provided by Dewesoft (historian.dewesoft.com). However, Dewesoft also offers the option to set up your own private server, which can run an MQTT broker, an InfluxDB database, and a Grafana Web Client.

In this part of the training, we'll learn how to set up and configure your own private server. Now, letâ€™s roll up our sleeves and turn our computer into our very own Historian server!

This part of the training requires:

1.) **Docker desktop application**. You can download it on the official **Docker website**.

Docker engine must be running all through this section of the training so make sure you run the Docker app and sign in.

2.) **Dewesoft Monitoring Docker package**. You can download it here.

Unzip the downloaded archive and **run install-windows.bat** if you're setting up the Historian server on a Windows machine, **or install-linux.sh** if you're setting it up on a Linux machine. Make sure to **read the README file**.

Thatâ€™s it, your server is up and running locally on your machine!

To access the Grafana interface of your local Historian instance, open your web browser and visit localhost:3000. If you wish to make your local Historian instance accessible over the internet, you will need to expose it by configuring the appropriate IP addresses, ports, and firewall settings. For detailed guidance, consult the Historian manual.
You can check which processes are running in the Docker application by either viewing them directly in the Docker desktop or by typing the â€œ**docker ps**â€ command into the Windows Command Prompt (CMD). This will display a list of all active Docker containers.

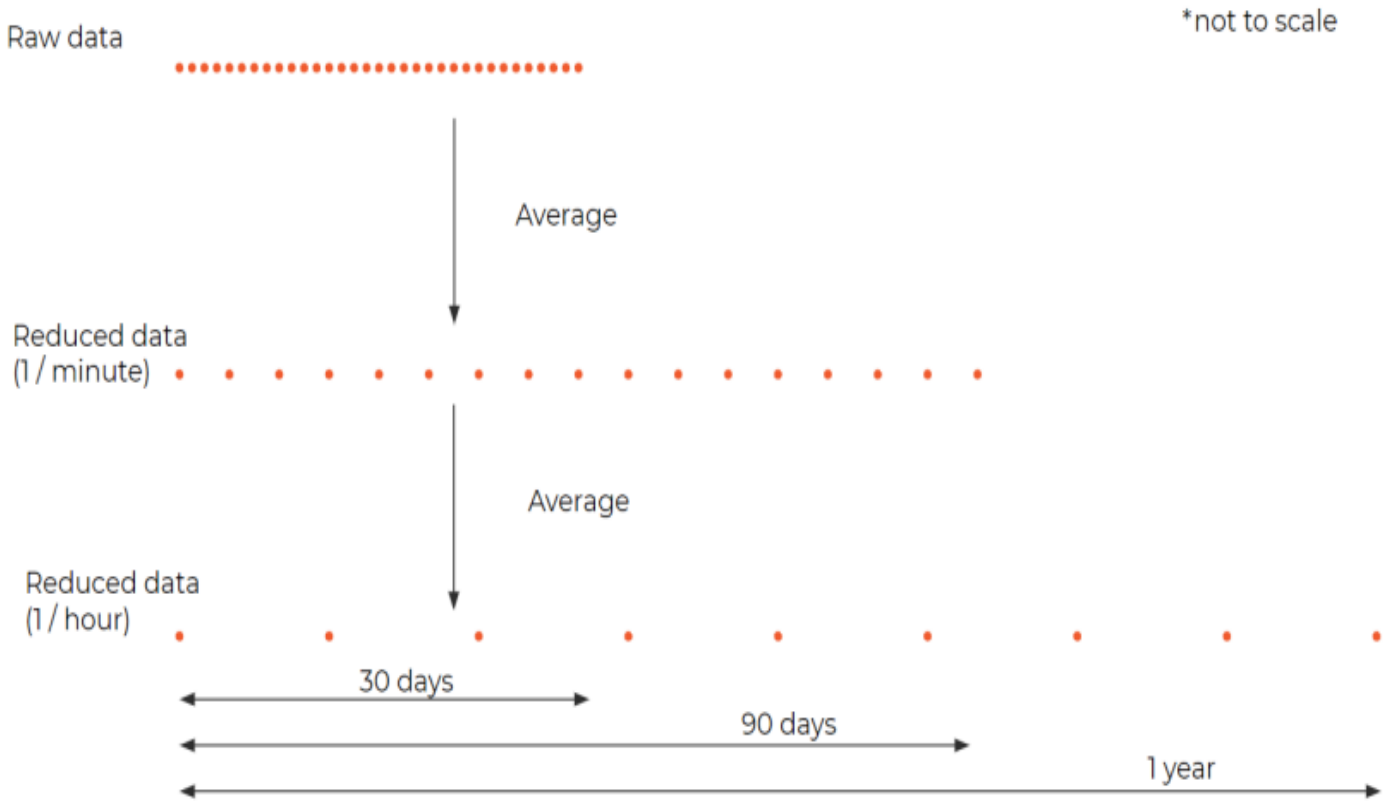
CHALLENGE: Setup Your Own Server

Congratulations for making it this far into the Dewesoft Historian training! By now, you should have a solid understanding of MQTT. Letâ€™s return to the section where we set up the subscriber and publisher in the DewesoftX Historian plugin. Your challenge now is to replace the public Historian broker with the one running in the Docker container on your local machine.

(Hint: Local addresses can be accessed using "localhost" instead of a URL or IP address.)

Data Retention Policies

Influx database includes an automatic procedure that can reduce the amount of data in the database after a certain amount of time. For example, the data can be originally uploaded to the database at 10 S/s, but after 1 month every 10 samples of data are averaged and the data rate is reduced to only 1 S/s. The figure below shows the data reduction graphically.



By default, the database of the Dewesoft Historian does not reduce any data, and all data is stored indefinitely. For instructions on how to enable different retention policies, please consult the Dewesoft Historian manual.

Grafana Web Client

Setup

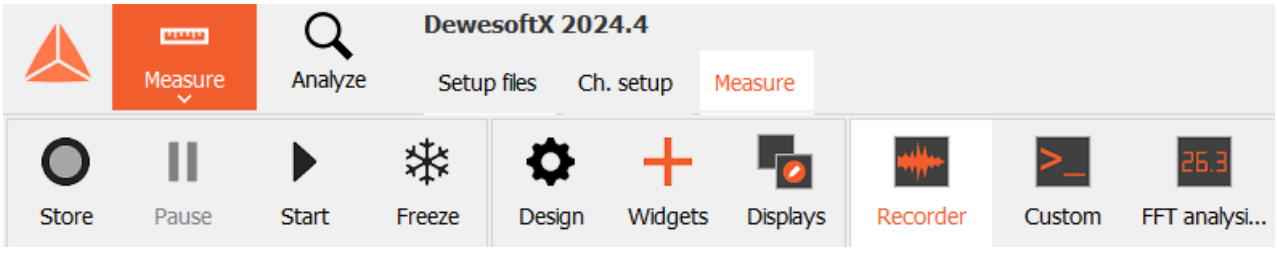
Grafana requires users to **set up an account** in order to log in. We saved this step for last since we couldn't test the public historian.dewesoft.com Grafana, as it's a public live system that requires a username and password provided by Dewesoft. However, since we now have a local server running that includes Grafana, we can proceed to test it locally.

We learned that processes running locally can be accessed using "localhost" instead of an IP address. Our local Grafana runs on port 3000, so to access it, open your browser and type localhost:3000. If the Docker process is running, the Grafana GUI should load, and you will be prompted to enter a username and password. By default, the credentials are "admin/admin", but for security reasons, you will be required to change these credentials upon first login.

Forgotten admin password cannot be recovered locally. If you forget the password, you will need to install Grafana again and you will lose the work done (configured dashboards, added users etc.)

Before setting up Grafana, we need to **ensure that the data channel is active, running, and being published** to our local broker. Refer to the â€œConfigurationâ€ section and set both the publisher and subscriber IP addresses to â€œlocalhostâ€. After changing the broker location from the Historian server to your local broker, verify that your simulated channel is being published.

To confirm your data is active, enter Measure Mode in DewesoftX. A graphical visualization of your active simulated channel should appear, indicating that everything is working correctly.

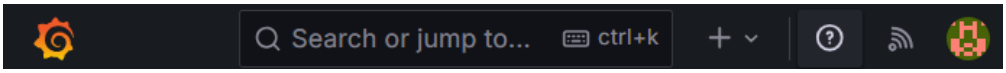


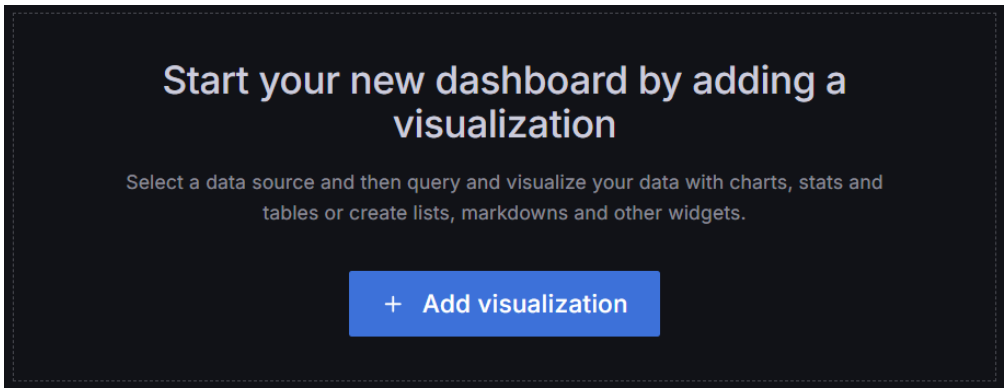
Leave this dewesoftX window open and switch to your browser.

Creating a dashboard

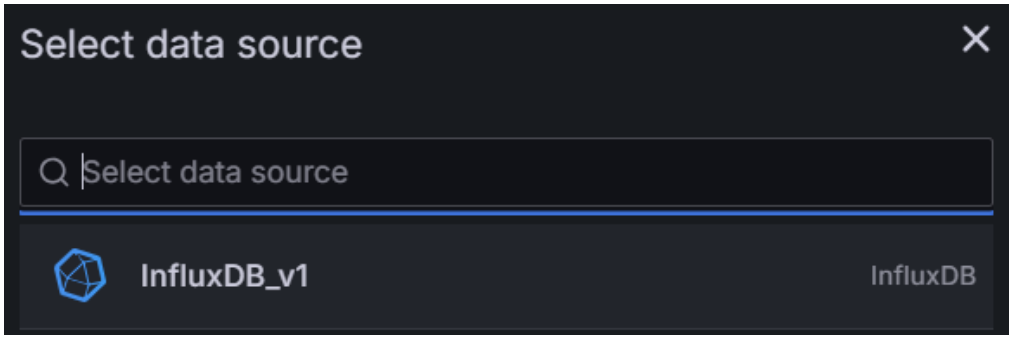
Grafana offers many possibilities of displaying and working with data. For a comprehensive guide, **Grafana official documentation** should be used.

To display the data a dashboard needs to be created and a panel added to it. A dashboard is added by clicking the plus button in the menu choosing option â€œnew Dashboardâ€ and selecting â€œ+ Add Visualizationâ€ in the screen that opens.

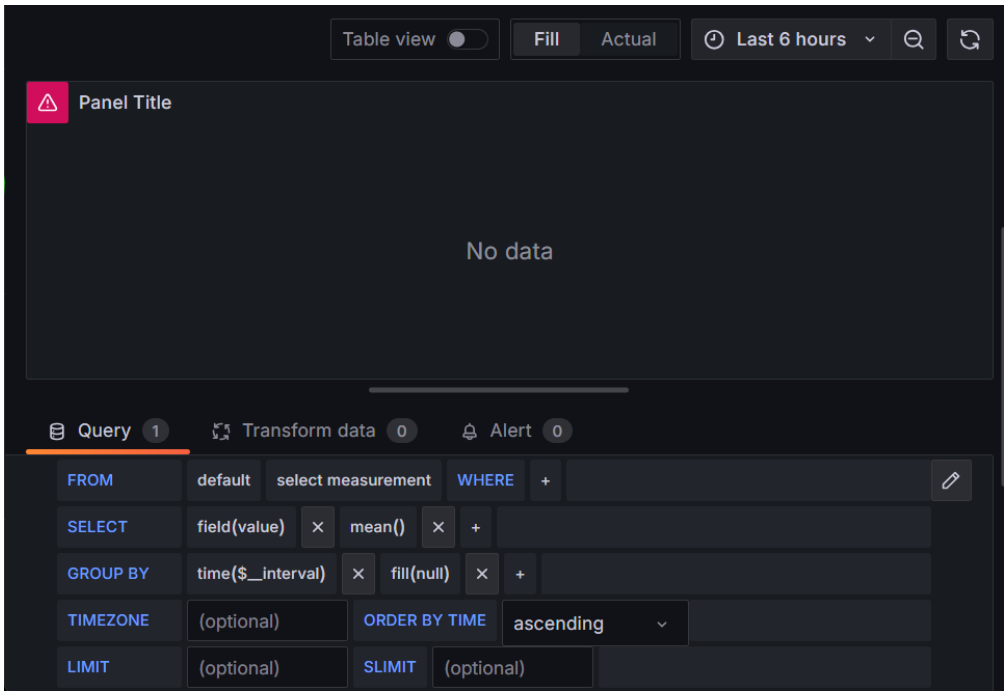




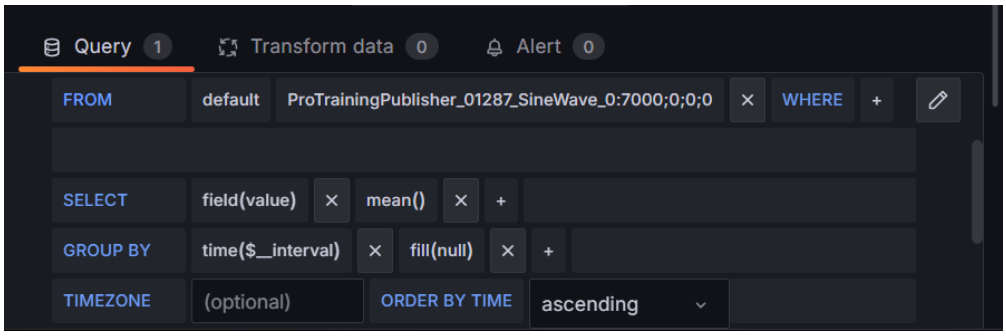
Data source (the address of the database) is already preset, you just need to select it:



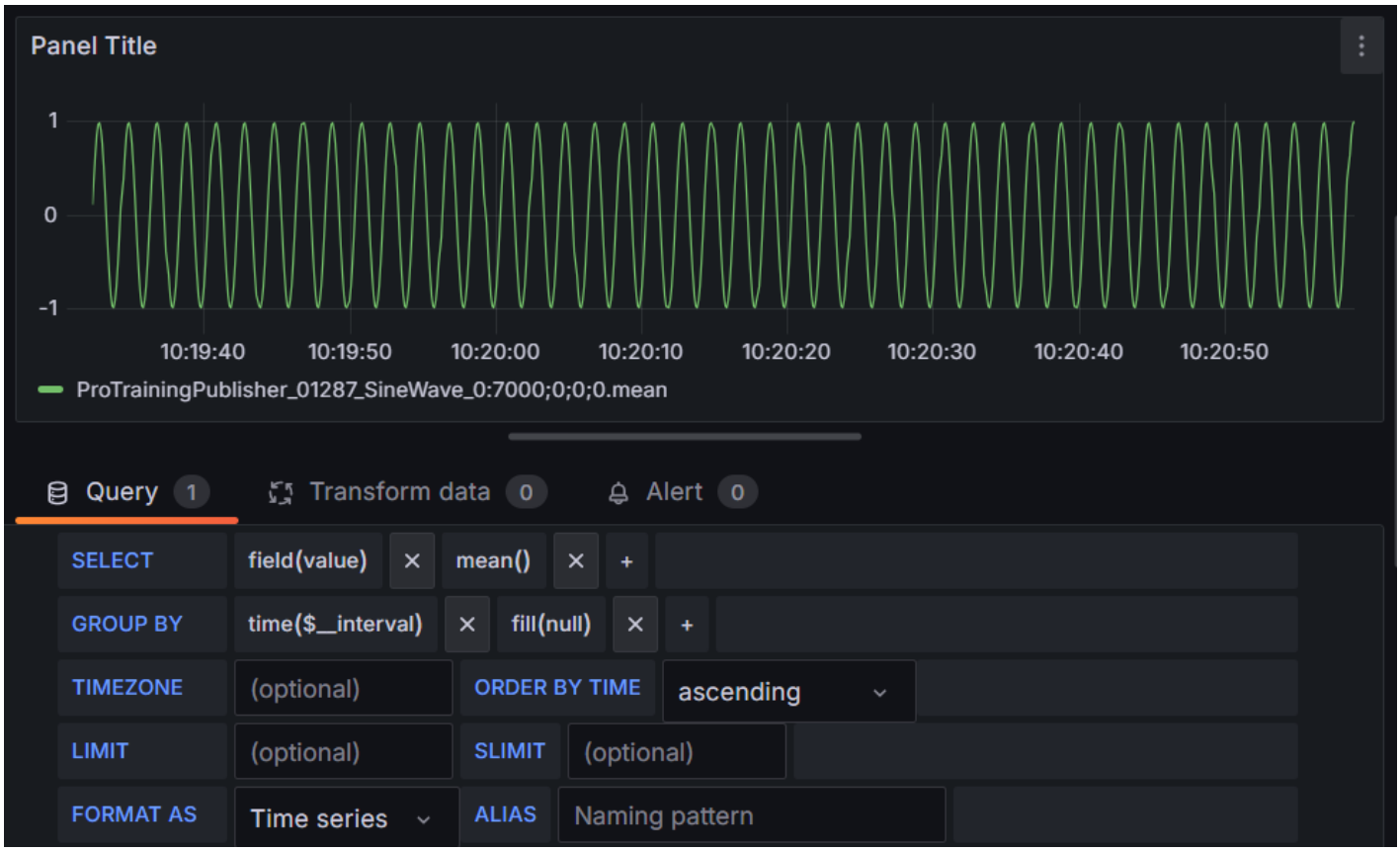
The following view appears:



Go to **Select Measurement**. A list of all channels ready for display will be shown. The database query is already preconfigured, so all that's left is to select the measurement channel. Your simulated channel should appear in the list.



After youâ€™ve configured the database query, your measurement should appear and should look something like this:

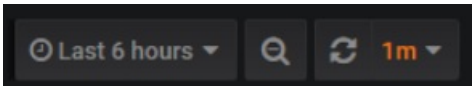


Grafana - Working with dashboards and panels

Grafana offers extensive options for visualizing and analyzing data. While its official documentation provides a detailed, comprehensive guide for working with Grafana, this manual will provide key tips specifically for handling measurement data typically used in Dewesoft Historian applications. These hints will help you get started with basic setups, dashboards, and visualization techniques, focusing on integrating Dewesoft's data efficiently into Grafana. For a deeper dive into Grafana's full capabilities, please refer to its official documentation.

Time axis

In Grafana, the time axis settings and screen refresh rate are controlled through three primary tools located in the upper right corner of the screen:

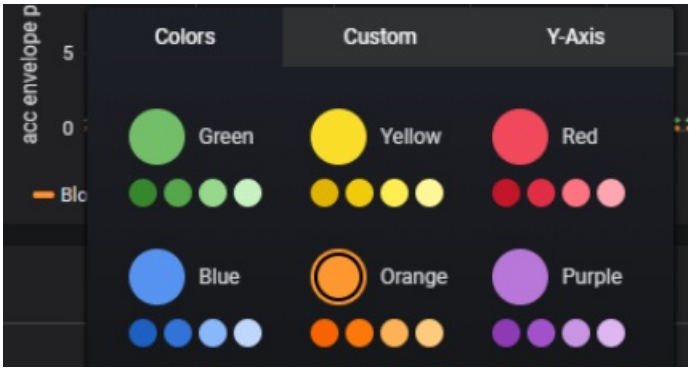


- **Refresh Button**:
  - The far-right button is used to manually refresh the screen. This can be useful when working with live data or making changes to the dashboard configuration. Additionally, the dropdown next to this button allows you to select the automatic refresh rate, ensuring the data updates periodically.
- **Time Interval**(X-axis):
  - The far-left button controls the time range displayed on the X-axis (e.g., "Last 5 minutes," "Last 1 hour," "Last 6 hours," etc.). This feature allows you to quickly switch between different time intervals to analyze data over varying periods.
- **Zooming**:
  - You can zoom in on the time axis by using a click-and-drag gesture directly on the data in the graph, similar to how it is done in Dewesoft software.
  - To zoom out, you can either click the middle button (reset zoom) or select a different time range using the far-left button.

Color

You can change the measurement color by:

- **Click on the colored line** next to the channel name in the legend at the bottom of the recorder display.
- A **color picker will appear**, allowing you to select from different preset colors.
- If you want to use a specific color, select the Custom tab from the color picker to input or select a custom color

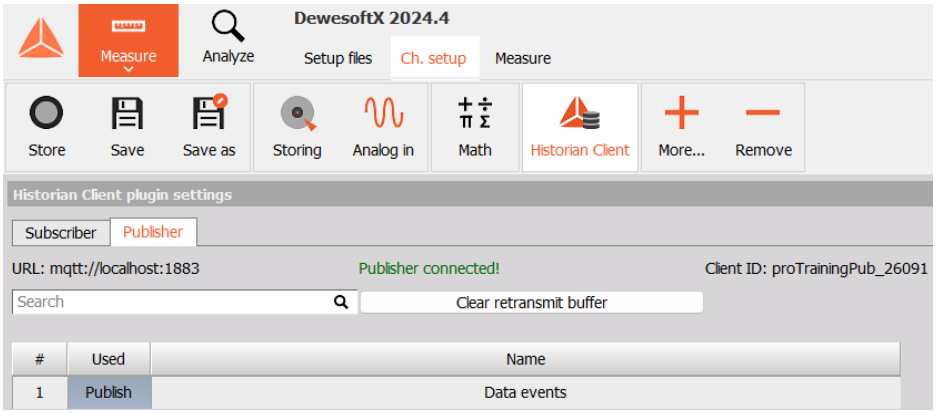

acc envelope p
5
0
Blo
Colors
Custom
Y-Axis
Green
Yellow
Red
Blue
Orange
Purple

Grafana - Annotations

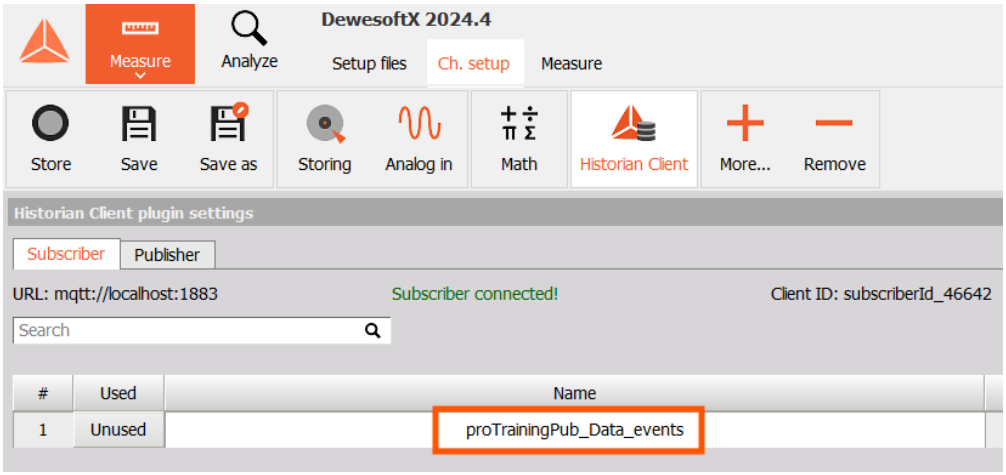
DewesoftX offers event functionality that allows you to set trigger points or conditions that activate events. A predefined example is the â€œStoringâ€ event, which automatically triggers whenever file storage starts or stops (custom triggers can also be created). These events can then be imported into Grafana and displayed as annotations.

To test Grafana event annotations, we will begin by configuring DewesoftX. Since the "**Storing**" event is predefined, we will use that as an example.

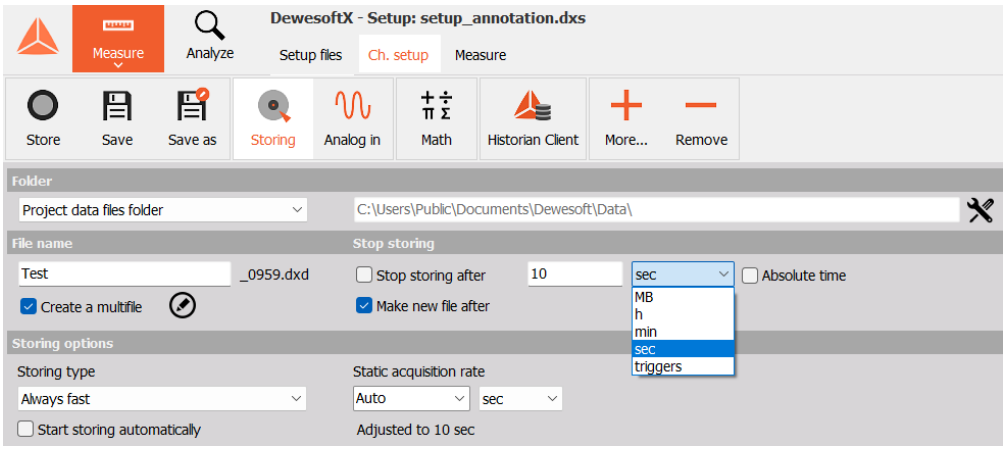
First, open DewesoftX and navigate to the **Historian Client Publisher**. Locate the Data_event channel and activate it by toggling the button in the "Used" column to "**Publish**."



Now switch to Subscriber section, find your Data events channel and remember the name of this channel for later:

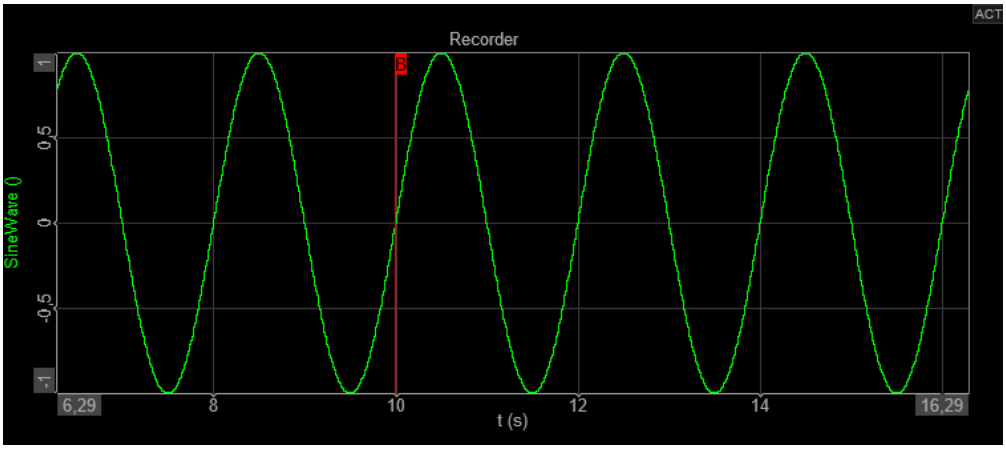


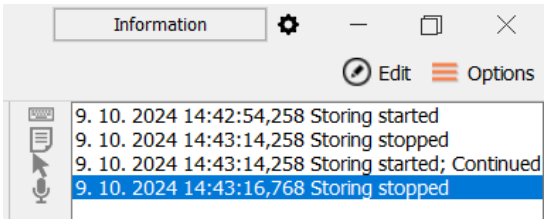
Now, go to **Channel Setup** and open the **Storing section**. Configure the storing functionality to create files sequentially by checking the â€œ**Create a Multifile**â€ option and selecting "**Make new file after**" checkbox. Enter the desired file saving interval in seconds (10 seconds or higher recommended), and ensure "**sec**" is selected as the unit from the dropdown menu.



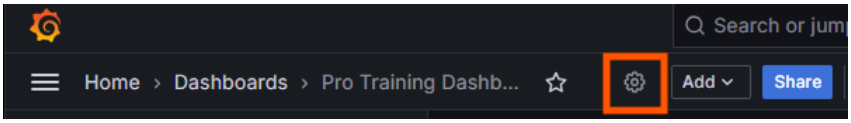
Once DewesoftX file storage is configured, click '**Store**' to begin the measurement process with file storage. DewesoftX will enter measurement mode, and if the storage configuration is correct, a red annotation will appear on the graph at every defined interval, indicating the occurrence of a new storage event.

Additionally, your events will be displayed in the **Event Viewer**, which is located in the top right corner of the interface.

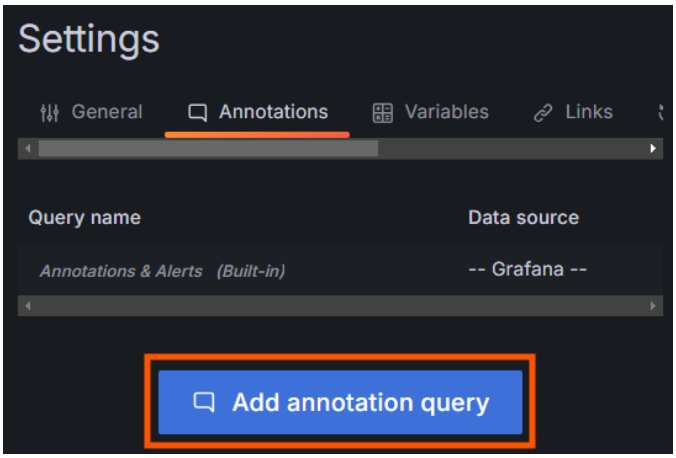




With your Data_event channel activated, storage properly configured, and DewesoftX running in measurement mode, switch to Grafana in your web browser. Find your Dashboard and open dashboard settings:



Go to the Annotations section and click "**Add annotation query**" to configure a new query that fetches and displays event data, such as the "**Storing**" events from DewesoftX.



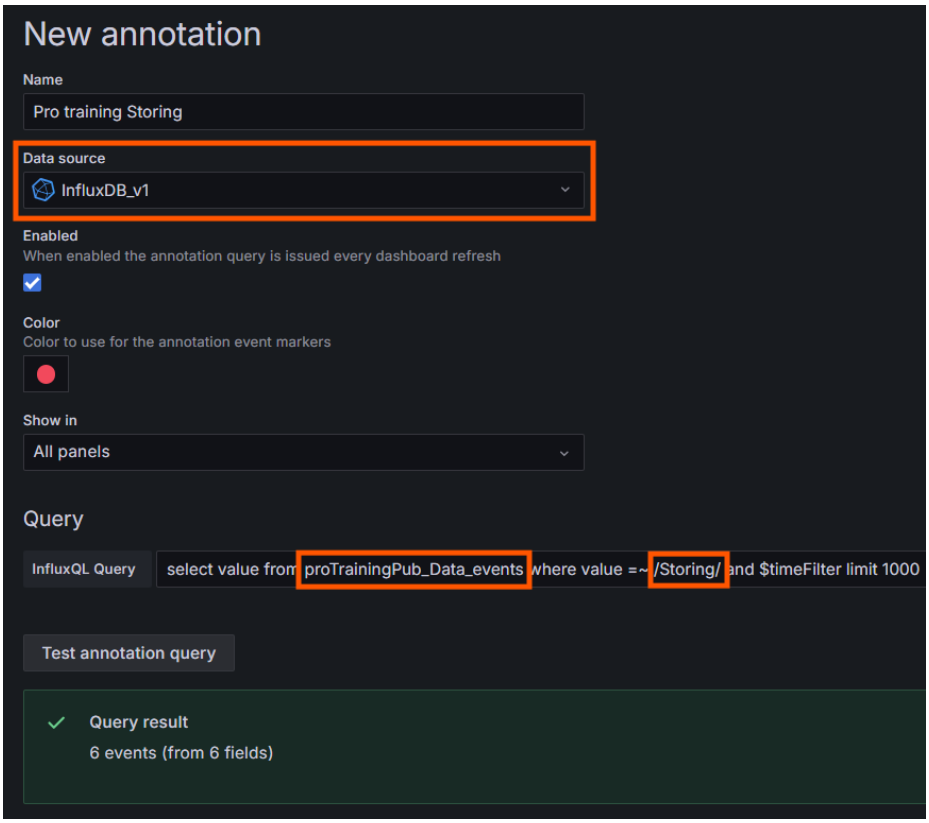
To properly configure annotations in Grafana using data from InfluxDB and events from DewesoftX, follow these steps:

- **Assign annotation name**:
  - Assign a descriptive annotation name that makes it easy to identify the specific event or data type being tracked.
- **Ensure the correct data source:**
  - Configure the data source to pull from **InfluxDB** instead of Grafana itself.
- **Enable annotation auto-refresh:**
  - Check the "**Enabled**" option to ensure that annotations are refreshed automatically with every dashboard refresh.

With the name of the Data_event publisher channel, which was activated earlier in DewesoftX, and the event name, such as "**Storing**" construct the following query:

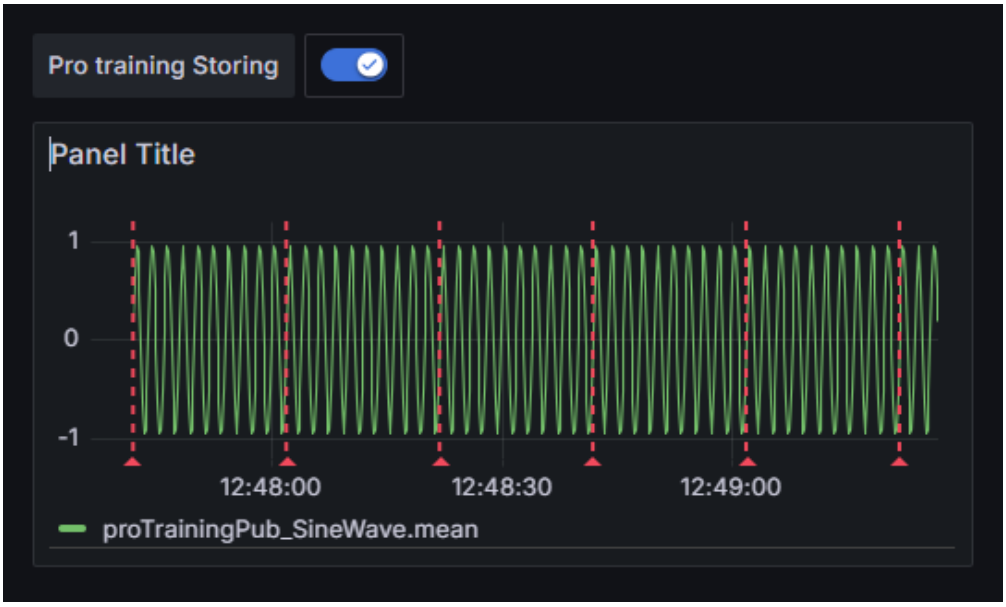
select value from **<Data_events publisher channel name>** where value =~ **/<event name>/** and $timeFilter limit 1000

If your Dewesoft measurement was running and you observed the "**Storing**" events occurring alongside the activated Data_events channel, those events have been saved in the database.



To verify if your events were successfully stored, click the "**Test annotation query**" button. If the query returns or finds your "Storing" events, you're ready to proceed. Apply the changes, save the dashboard, and return to the corresponding graph.

A switch should appear at the top of your graph, labeled with the name of your annotation. **Ensure that the time range of your graph matches the period when the "Storing" event occurred**. Finally, turn the switch on to display your annotations on the graph in the color you specified during setup.

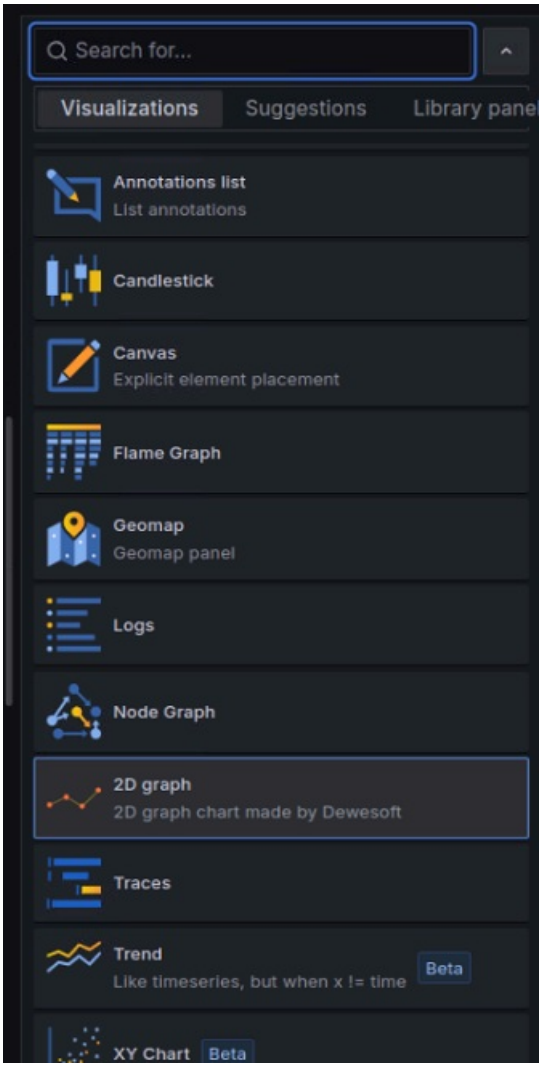


Grafana 2D Graph

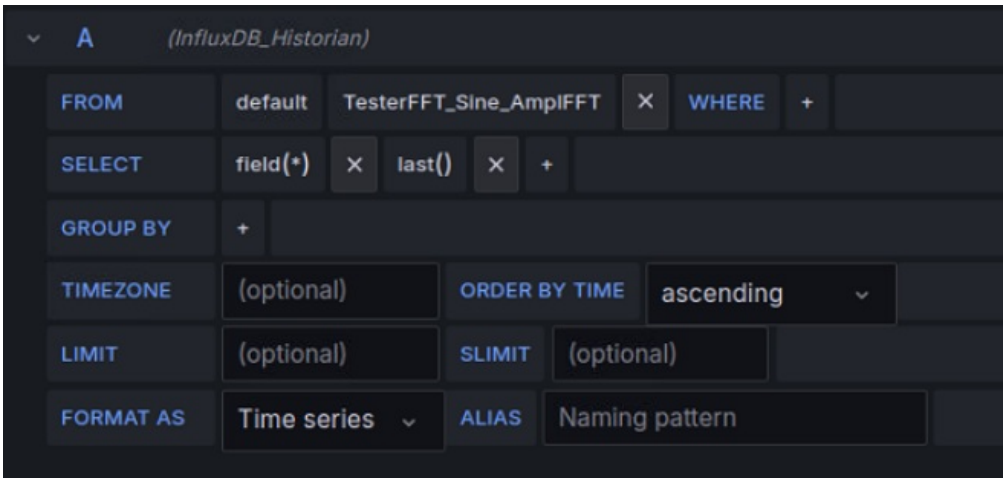
The Grafana 2D graph plugin, developed by Dewesoft, allows the display of vector channels in Grafana. Common vector channels include FFTs and Scope Math channels. This graph serves a similar purpose to the 2D graph in DewesoftX software, offering a familiar visualization tool for analyzing data.

The Grafana 2D graph plugin comes pre-installed in the Dewesoft Historian Docker package, making it ready to use without additional setup.

To set up a 2D graph in Grafana, start by adding a new panel and selecting the 2D graph plugin as the visualization type. Alternatively, you can use the preconfigured Default Dashboard that comes with the package, which already has the correct plugin and query selected. To use this, simply duplicate the dashboard and choose the channel name configured in DewesoftX Historian Client publisher, with the suffix _AmplFFT.



Assign one of the vector channels from the database with the suffix _AmplFFT and set up the query exactly like shown in the picture below.



If the query is set correctly, the 2D graph plugin visualization should appear similar to the example shown below.

