

intdash Tutorial 1: Uploading a CSV File and Visualizing Data (For intdash All-in-One Users)

intdash All-in-One Version 202209

7th edition (November 2022)

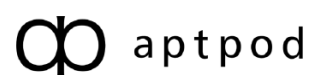


Table of contents

01 Preparation	4
1.1 Set up intdash environment	4
1.2 Create an edge account	4
1.3 intdash SDK	4
1.4 Prepare Jupyter Notebook runtime environment.....	4
02 Tutorial 1A: Visualize the sample data	5
2.1 Create a dashboard in Data Visualizer	5
2.2 Upload a CSV file.....	8
2.3 Visualize uploaded data on dashboard.....	10
03 Tutorial 1B: Create your own dashboard	13
3.1 Check the content of the CSV file to be uploaded	13
3.2 Create data settings	14
3.3 Place visual parts	17
3.4 Upload CSV file.....	21
3.5 Visualize uploaded data on dashboard.....	21
04 Tutorial 1C: Process server data using intdash SDK	22
4.1 Change dashboards.....	22
4.2 Upload the CSV file.....	23
4.3 Retrieve and process data	23
4.4 Display processed data.....	23
05 Tutorial 1D: Upload a large number of CSV files using intdash SDK	24

This tutorial will show you how to upload CSV files to intdash and visualize them using Visual M2M Data Visualizer (hereafter referred to as Data Visualizer).

By reading this tutorial, you will learn how to upload CSV files, how to create dashboards in Data Visualizer, and how to retrieve data from the server and process data using the intdash SDK for Python (hereafter referred to as intdash SDK).

1. Upload a sample CSV file and visualize the data in Data Visualizer.
→ [Tutorial 1A: Visualize the sample data](#) (p. 5)
2. Build an original dashboard on Data Visualizer to visualize CSV data.
→ [Tutorial 1B: Create your own dashboard](#) (p. 13)
3. Process the data stored on the intdash server using intdash SDK and visualize it with Data Visualizer.
→ [Tutorial 1C: Process server data using intdash SDK](#) (p. 22)
4. Upload a large number of CSV files using the intdash SDK.
→ [Tutorial 1D: Upload a large number of CSV files using intdash SDK](#) (p. 24)

Important:

- Specifications in this document are subject to change without notice. This document is for informational purposes only and does not guarantee the specifications.
- The screens used in the explanation are examples. Depending on your environment and application version, some of the displays and procedures may differ.

Note: Company names, service names, product names, etc. described in this manual are registered trademarks or trademarks of each company. "™" and "®" are not specified in the text and figures.

Attention: This document has been translated using machine translation services and may contain inaccuracies and translation errors. Please also refer to the official version in Japanese.

01 Preparation

To run tutorials 1A through 1D in this document, you need to prepare the following.

- Set up intdash environment
- Create an edge account
- Install intdash SDK for Python
- Prepare Jupyter Notebook runtime environment

1.1 Set up intdash environment

Refer to [Setting up intdash server using AMI](#) to set up intdash environment.

1.2 Create an edge account

In intdash, time series data is managed as "data from the edge". Therefore, even when importing a CSV file into intdash, it is necessary to specify "from which edge the data was obtained (data acquisition source)".

Subsequent tutorials will treat data points in the CSV file as the data retrieved by `edge1`. Therefore, create an edge account called "edge1" by following the steps below.

1. Login to intdash with your username and password, and open My Page (https://<hostname_of_intdash_server>/users/me/).
2. Click [Edges] > [Create Edge] and create an edge account named `edge1`.

You will see the UUID and the client secret, but you do not need to copy them for this tutorial scenario.

1.3 intdash SDK

To run [Tutorial 1C: Process server data using intdash SDK](#) (p. 22) and [Tutorial 1D: Upload a large number of CSV files using intdash SDK](#) (p. 24), use a developer tool called intdash SDK for Python. Install intdash SDK for Python by referring to the [SDK documentation](#).

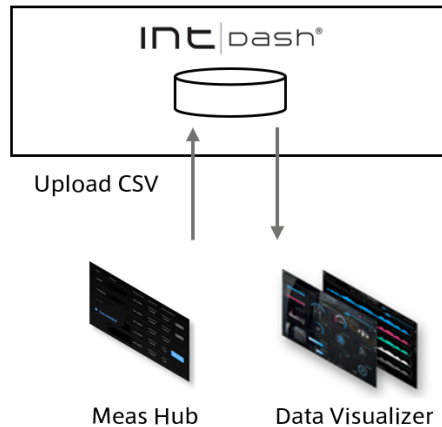
1.4 Prepare Jupyter Notebook runtime environment

In this tutorial, Jupyter Notebook is used as the execution environment of Python.

1. Clone the Jupyter Notebook tutorial repository [aws-marketplace-tutorials](#).
2. Follow the README to set up a Jupyter Notebook runtime environment.

02 Tutorial 1A: Visualize the sample data

In the first tutorial, we will upload a sample CSV file and visualize it with Data Visualizer.



2.1 Create a dashboard in Data Visualizer

First, create a dashboard on Data Visualizer.

2.1.1 Open Data Visualizer

Note: The URL scheme (http or https), host name, and password differ depending on the configuration. If you do not have these, check with the person in charge of the server setup.

1. Using the Chrome browser, go to `https://<hostname_of_intdash_server>/`. The sign-in screen will be displayed. Sign in with your user name and password.

Once you sign in, the Project Console application will appear.

2. Since we will use "Global Project" here, click on [Details] of "Global Project" in the list of projects.

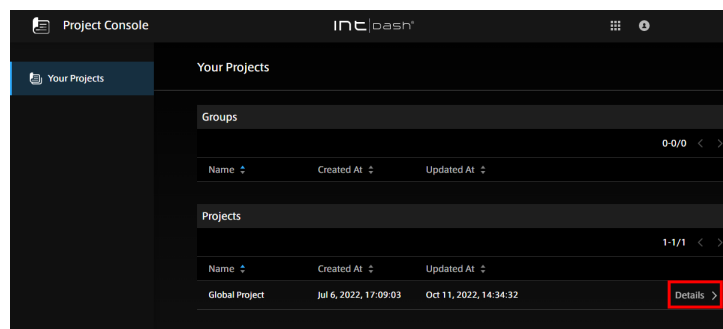


Fig. 1 Select Global Project

3. From the list of applications, select Data Visualizer.

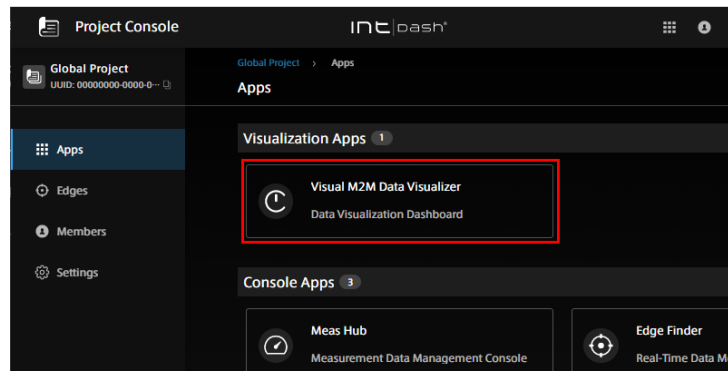


Fig. 2 Open Data Visualizer

The Data Visualizer dashboard will appear.




Fig. 3 Data Visualizer dashboard

2.1.2 Import a data settings file

Set how to parse data retrieved from intdash. In this tutorial, load a pre-prepared data settings file (called a DAT file because its extension is ".dat") to set up all data settings.

1. Download the following DAT file.

[1_1_csv_parse.dat](#)

2. Click [Data Settings] () on the left side of the screen, and then click [Import].

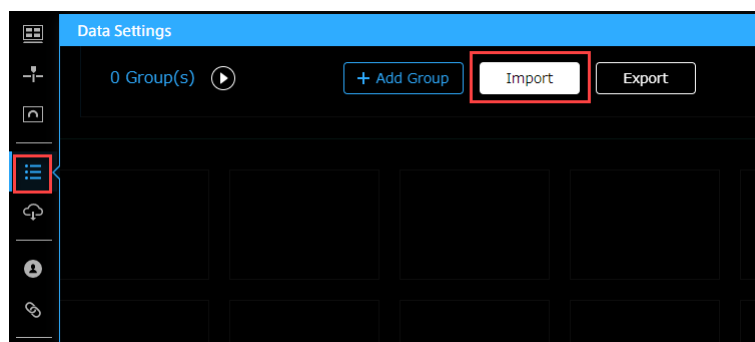


Fig. 4 Importing data settings files


3. In the file selection dialog, select `1_1_csv_parse.dat` downloaded in step 1 and import it.

2.1.3 Load a dashboard configuration file

Finally, set up a dashboard for visualization. In this tutorial, load a pre-prepared dashboard configuration file (called a Screen file).

1. Download the following Screen file.

[1_1_csv_screen.scrn](#)

2. Click the [Screen list] () button at the bottom of the screen, and then click [Import].

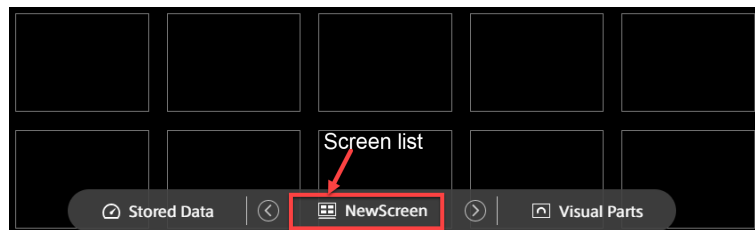


Fig. 5 Screen button

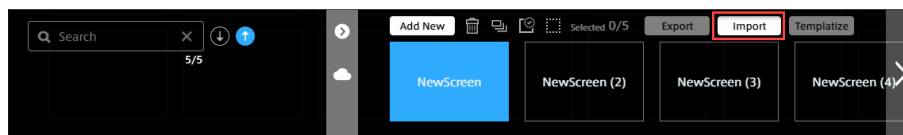



Fig. 6 [Import] button

3. Select and import the `1_1_csv_screen.scrn` downloaded in step 1.
4. Select the imported [CSV Screen].
5. Set each panel on the dashboard to display the data from `edge1`.

The imported dashboard is set to display data from "Unknown Edge". Click  (Replace Edge) to replace the source edge setting "Unknown Edge" with `edge1`.

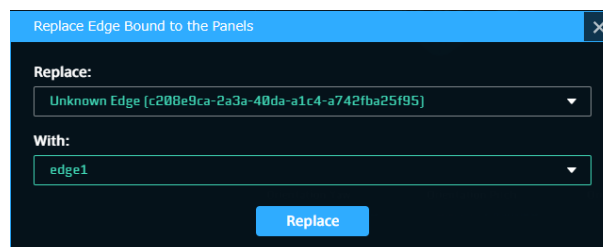


Fig. 7 Replace "Unknown Edge" with `edge1`

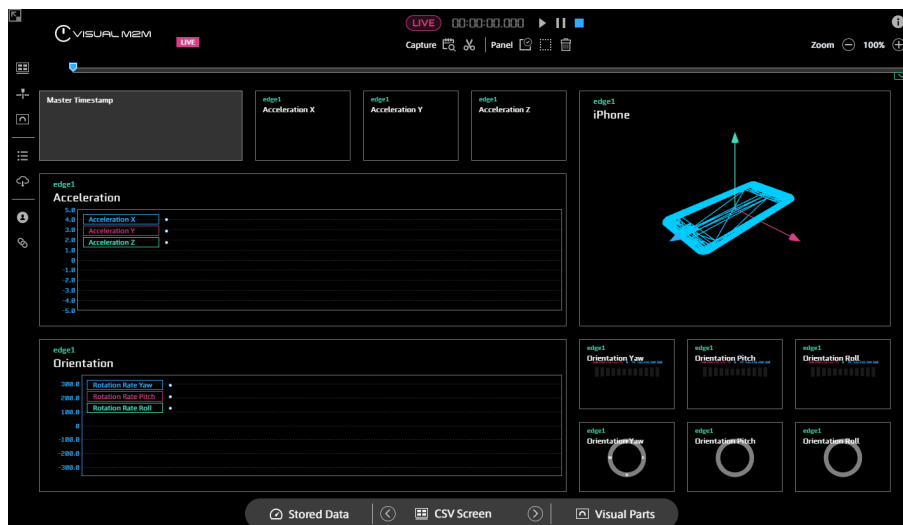


Fig. 8 Dashboard screen

This completes the data settings and the dashboard settings on the Data Visualizer.

2.2 Upload a CSV file

In the previous section, the dashboard was completed. Next, upload the CSV data you want to visualize.

2.2.1 Check the contents of the CSV file to be uploaded

First, check the CSV data you want to visualize. In this chapter, the following CSV file is used as a sample.

[sample.csv](#)

The sample file contains sensor signal values obtained from a 6-axis sensor. Timestamps are stored in the first column, and signal values are stored as numerical data in the following columns. The first row is a header row that contains the name of data.

Note: Use UTF-8 as the encoding of the CSV file. If you use an encoding other than UTF-8, multi-byte characters such as Japanese may be garbled.

2.2.2 Open the Meas Hub application

Return to the Project Console tab in your web browser and open the Meas Hub application.

2.2.3 Upload a CSV file

1. Click [Measurement Uploads] on the side menu to move to the CSV upload screen.
2. Click [Create Measurement from CSV] and select "edge1".

With this setting, the uploaded CSV file will be edge1 data.

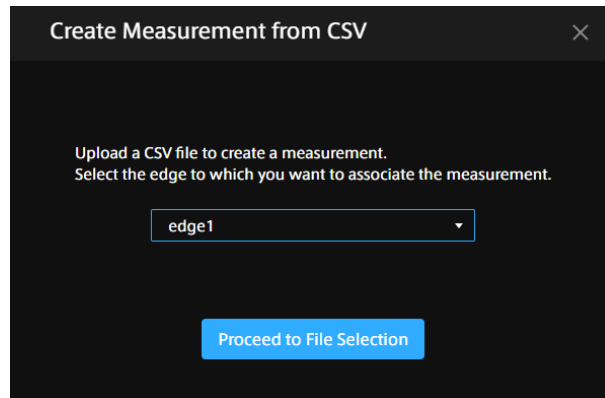


Fig. 9 Select an edge

3. Click [Proceed to File Selection], select the sample file, and upload it.

When the CSV file has been sent successfully, the file will be displayed in the file list.

4. When the CSV file is uploaded to intdash, the data registration process is executed in the background. Please wait for a while until the status of the uploaded file becomes "Succeeded".

Note: If you upload files with the same name multiple times on the same day, a random string will be added to the beginning of the second and subsequent file names (e.g. WXeiMUYkBU_sample.csv).

5. When the status becomes "Succeeded", copy the UUID of the measurement ("Created Measurement").

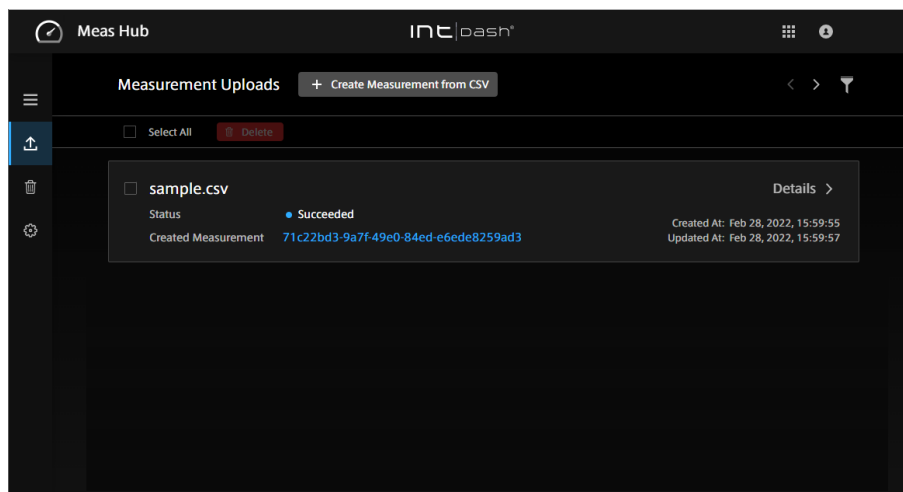


Fig. 10 List of uploaded files

A single CSV file is converted into a single measurement and stored in intdash. A measurement is a set of data from the start of the measurement to the end of the measurement.

2.3 Visualize uploaded data on dashboard

With the steps so far, the dashboard and the CSV file are ready. Finally, visualize the uploaded data using the dashboard.

2.3.1 Select uploaded data from Stored Data

1. Return to the Data Visualizer tab in your web browser.
2. Click [Stored Data] at the bottom of the screen, then click [Measurements]. A list of measurements is displayed.

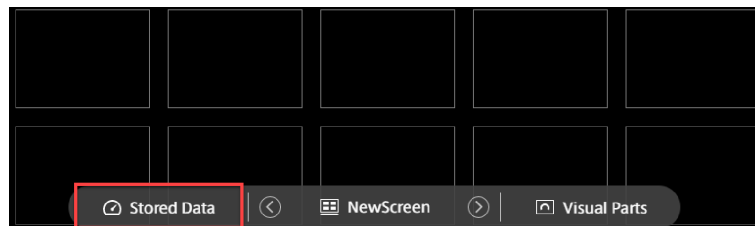


Fig. 11 List of measurements

3. Since the timestamp of the uploaded data `sample.csv` is July 2020, specify the date range as shown in the figure below to display the data for July 2020.

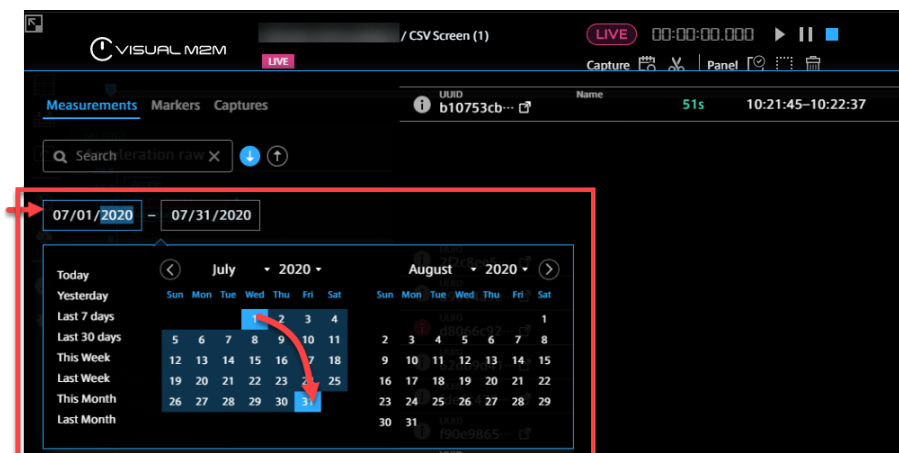


Fig. 12 Drag the date range to view data for July 2020

4. Select the measurement with the UUID you copied. The data will start loading immediately and will play back on the dashboard after a while.



Fig. 13 Dashboard screen during playback

2.3.2 Basic operation (play, pause, move playback position)

You can play or pause the data from the control panel at the top of the screen. You can also move the playback position during pause by using [<<] and [>>] in the control panel. The playback speed can be adjusted with the [Speed] drop-down list.



Fig. 14 Control panel

2.3.3 Advanced usage (changing visual part settings)

In Data Visualizer, you can change the visualization by changing the setting of visualization parts (hereinafter referred to as visual parts) on the dashboard. You can configure a visual part on the [Panel Option] tab. (However, simple visual parts such as Value Current, Text Stream, do not have [Panel Option].)

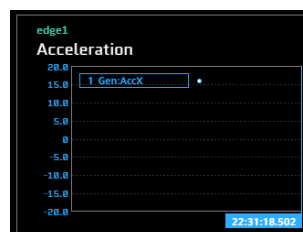


Fig. 15 An example of a visual part

1. Right-click on the displayed visual part to open the [Panel Settings] tab.
2. Open the [Panel Option] tab. Change the setting of the visual part to your liking.
For example, the [Panel Option] tab of a Line Graph includes the following items.

- Data Range: Specify the vertical display range
- Time (X) Interval (sec): Specify the tick interval for the horizontal axis.
- Show Points: Show markers on the line chart (The number of markers displayed on the screen is adjusted automatically.)

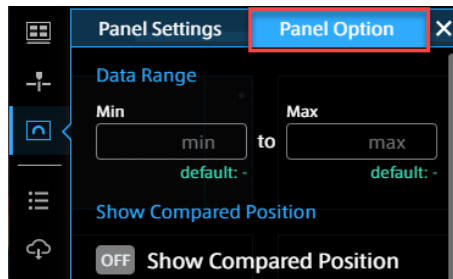


Fig. 16 Example: [Panel Option] tab of Line Graph

2.3.4 More advanced usage

This tutorial introduces only a few of the rich features of Data Visualizer. For more detailed usage, see [Data Visualizer operation manual](#).

Also, see the following chapters for other use cases.

- If you would like to create your own dashboard, see [Tutorial 1B: Create your own dashboard](#) (p. 13).
- If you would like to process the uploaded data using intdash SDK, see [Tutorial 1C: Process server data using intdash SDK](#) (p. 22).

03 Tutorial 1B: Create your own dashboard

In the second tutorial, we will build a new dashboard on Data Visualizer by placing visual parts and bind data.

3.1 Check the content of the CSV file to be uploaded

First, check the format of the CSV file you want to visualize. In this tutorial, use the same CSV file used in [Tutorial 1A](#) (p. 5). See [Check the contents of the CSV file to be uploaded](#) (p. 8) for the content of the file.

3.1.1 Uploadable CSV file format

The CSV file uploaded to intdash must meet the following conditions.

- The first line (header line) must be the names of each column as string values.
- The first column (timestamp column) must store timestamps
 - The header of the timestamp column must be `time`.
 - Timestamp format must conform to RFC 3339
- From the second row and the second column onwards, the cells must contain data (data area).
 - Data must be stored in the data area in the form of numbers or strings

3.1.2 Contents of sample CSV file

In the sample CSV file, numerical data `sp_ACCX` is stored in the second column, numerical data `sp_ACCY` is stored in the third column. You can see that there are six types of time series data stored in the 2nd to 7th columns of the file.

Header for timestamps: "time"

	A	B	C	D	E	F	G
1	time	sp_ACCX	sp_ACCY	sp_ACCZ	sp_Yaw	sp_Pitch	sp_Roll
2	2020-07-17T01:21:45.911747000Z	0.038937	0.491985	0.205379	-2.33905	1.28591	-2.08293
3	2020-07-17T01:21:45.931913000Z	0.139753	0.429696	-0.064305	-1.5008	-0.25512	-0.36002
4	2020-07-17T01:21:45.952079000Z	0.102823	0.398769	-0.076985	-0.92571	-0.6008	0.26643
5	2020-07-17T01:21:45.972245000Z	0.046802	0.320168	-0.009198	-1.03047	0.57966	0.15936
6	2020-07-17T01:21:45.992411000Z	0.00981	0.294638	-0.049083	-1.42192	-0.23358	0.40282
7	2020-07-17T01:21:46.012577000Z	0.05328	0.288342	-0.089622	-1.31317	0.73032	1.06981
8	2020-07-17T01:21:46.032743000Z	0.026842	0.302396	-0.235675	-0.26605	-1.1092	1.83864
9	2020-07-17T01:21:46.052908000Z	-0.027949	0.384417	-0.165986	1.11376	-0.27279	1.49542
10	2020-07-17T01:21:46.073074000Z	-0.064386	0.429939	-0.112762	2.09998	0.56316	0.67015
11	2020-07-17T01:21:46.093240000Z	-0.06718	0.441308	-0.019577	1.91861	-0.32435	0.74318
12	2020-07-17T01:21:46.113406000Z	-0.098304	0.435567	0.105435	0.19692	-0.22735	0.48849
13	2020-07-17T01:21:46.133572000Z	-0.098501	0.455999	0.061387	-1.00937	-0.07206	0.70936
14	2020-07-17T01:21:46.153737000Z	-0.109028	0.524762	-0.031624	-0.94903	-0.32082	0.45419
15	2020-07-17T01:21:46.173904000Z	-0.117208	0.513612	-0.081992	-0.54198	-0.04727	-0.18997
16	2020-07-17T01:21:46.194070000Z	-0.145731	0.456545	-0.053064	-0.29041	0.41748	-1.48417

Timestamps (RFC 3339)

Fig. 17 Contents of CSV file

3.2 Create data settings

Next, create data settings suitable for the content of the CSV file.

3.2.1 How CSV files are stored in intdash

Before creating data settings, this section explains how the data is stored in intdash.

- The data in each cell in the data area of the CSV file is given a data name and a timestamp, and is stored in intdash as one data point. For example, in the case of the following CSV file, 3 types of data exist in 4 timestamps, so a total of 12 data points are stored in intdash.

	A	B	C	D
1	time	DATA_1	DATA_2	DATA_3
2	2020-07-28T09:33:18.846165000Z	-0.691558	-0.169586	0.671237
3	2020-07-28T09:33:18.866331000Z	-0.292441	-0.026096	0.85781
4	2020-07-28T09:33:18.886497000Z	-0.011041	-0.050918	0.846662
5	2020-07-28T09:33:18.906663000Z	0.038279	-0.016691	0.565293

Fig. 18 Example CSV file containing 12 data points

- Type information is assigned to each data point according to the data. For example, data written as an integer in a CSV file is given a type of integer, data written as a decimal value is given a type of floating-point number, and data written as a character string is given a type of character string.
- intdash assigns an identification number called a channel number in addition to the above information. The CSV upload feature automatically assigns a channel number 1.
- When Data Visualizer retrieves data from intdash, it requests data with three pieces of information other than the timestamp (data name, data type and channel) and a range of timestamps.
- In data settings, you need to specify three pieces of information (data name, data type, channel) to identify data.

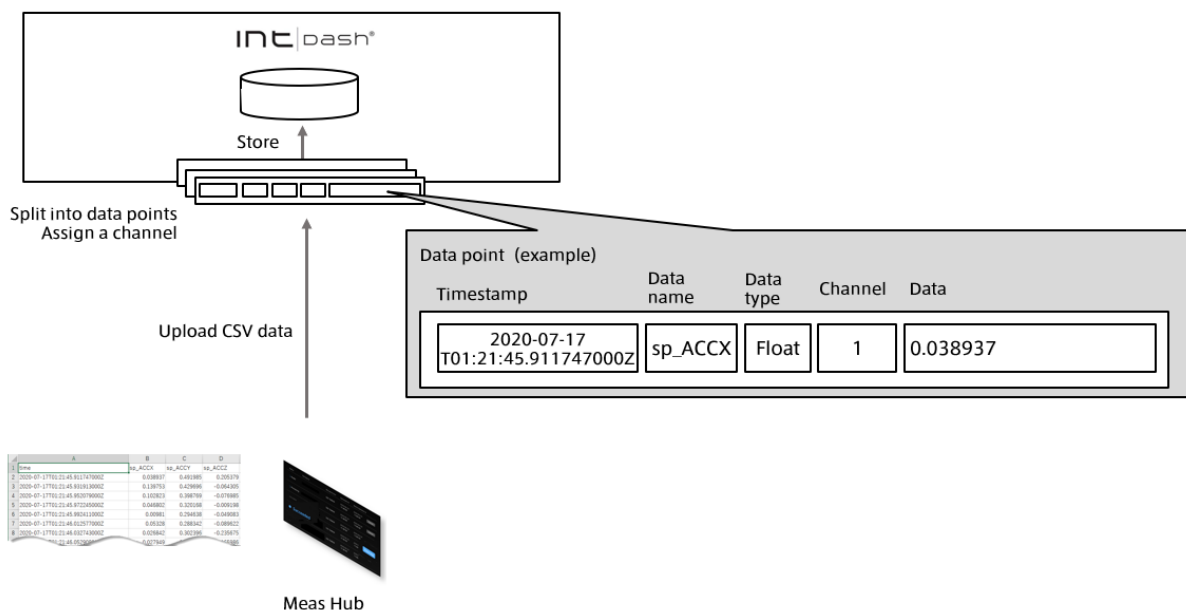


Fig. 19 CSV data upload with Meas Hub

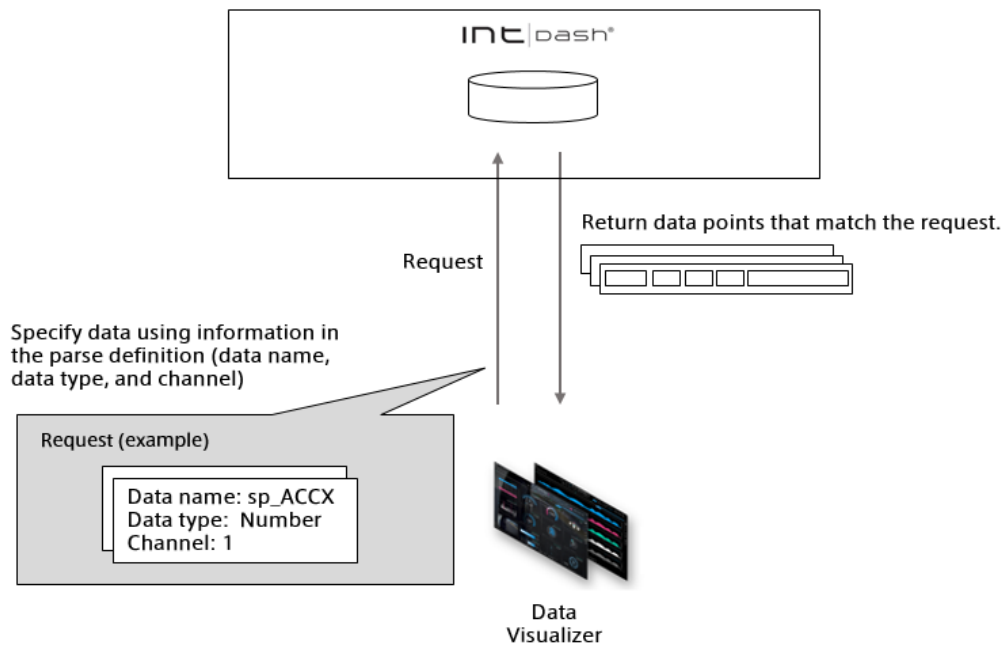



Fig. 20 Display data with Data Visualizer

In the following sections, we will add data definitions corresponding to each column of the sample CSV file.

3.2.2 Add a data settings group

Click [Data Settings] () located on the left side of the screen to open the data settings management window. In this window, multiple data settings are grouped together. First, click [Add Group] to create a group. An empty group named "New Data Group" is created.

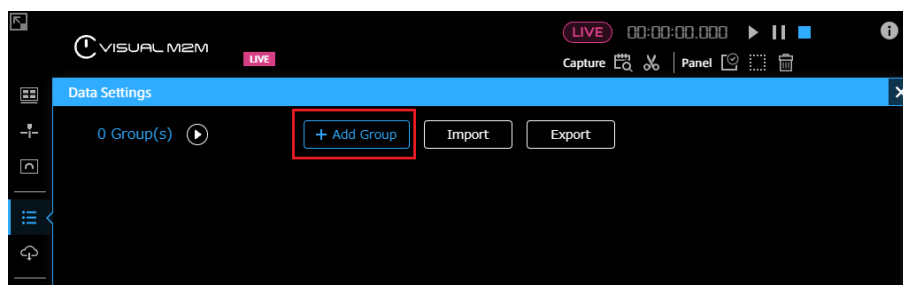


Fig. 21 Add a group

3.2.3 Specify a data type and a parsing type

The following instructions use the `sp_ACCX` column among the six columns of the CSV file as an example.

1. In the New Data Group, click  and click [Add Data].

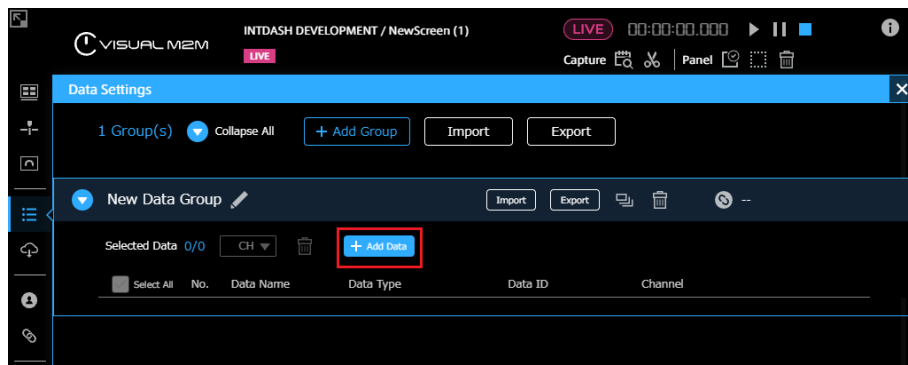


Fig. 22 Add data

2. Configure as follows.

- Data Name: acc_sensor_x
- Display Name: ACC (x-axis)
- Target Data:
 - Data Type: Float
 - Data ID: sp_ACCX
 - Channel: 1
- Conversion Settings:
 - Conversion Type: No Conversion (As Number)
- Display Settings:
 - Display Type: Number
 - Range: $-5 \leq \text{Value} \leq 5$
 - No. of Digits after Decimal Point: 2

Fig. 23 Data settings for sp_ACCX

This completes the data settings for the sp_ACCX column. Register data settings for the other 5 columns in the same way.


Note: Since only character strings or numbers can be contained in a CSV file, the CSV upload feature can only handle character string, integer, and floating point number. However, intdash itself can also handle binary format messages such as CAN (Controller Area Network) and MAVLink.

To visualize values stored in a binary format message, the message must be parsed. The parsing types other than [No conversion] are used for such message data. For details, refer to [Data Visualizer operation manual](#).

3.3 Place visual parts

Next, place the visual parts to make a dashboard.

3.3.1 Create a new dashboard

First, prepare an empty dashboard. Click [Screen list] () button at the bottom of the screen, then click [Add New]. This will create a new dashboard.

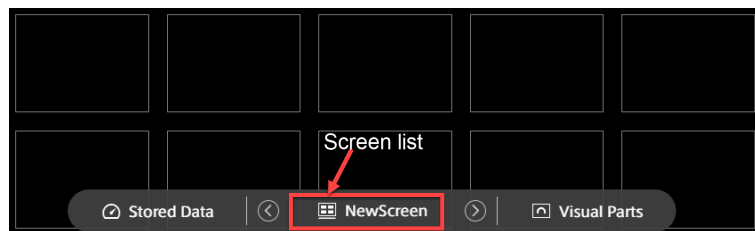


Fig. 24 Screen menu

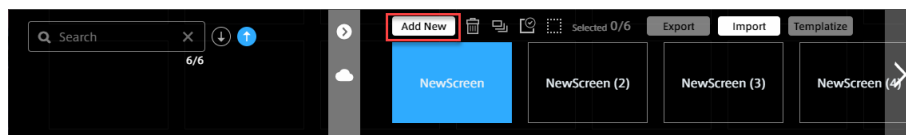


Fig. 25 [Add New] button

3.3.2 Reserve spaces for visual parts

Then click on any frame in the created dashboard and click on the button labeled "2 x 2". This will reserve 2x2 squares of space for one visual part. When the space is reserved, the [Panel Settings] tab opens on the left side of the screen. This window is used for visual part settings.

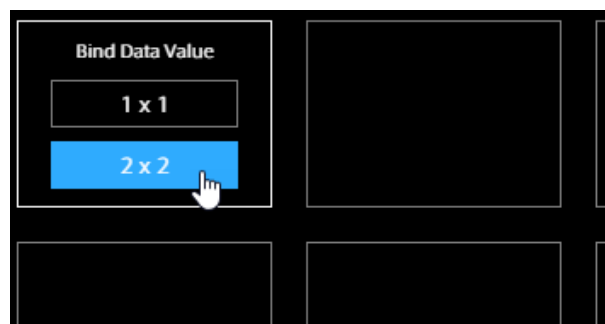


Fig. 26 [2 x 2] button

3.3.3 Set the visualization method

Once you have reserved space for the visual part, you can then specify how to visualize data.

1. On the [Panel Settings] tab, click the button under the "Visual Parts" heading.

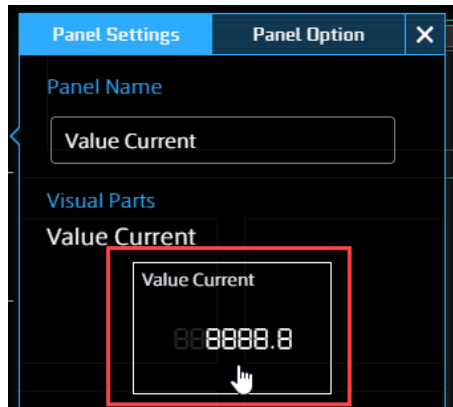


Fig. 27 Specify a visual part

A window will open displaying a catalog of visualizations.

2. From the catalog displayed, you can select any visualization representation that suits your visualization application. In this example, select [Line Graph] in the [Graph] group.

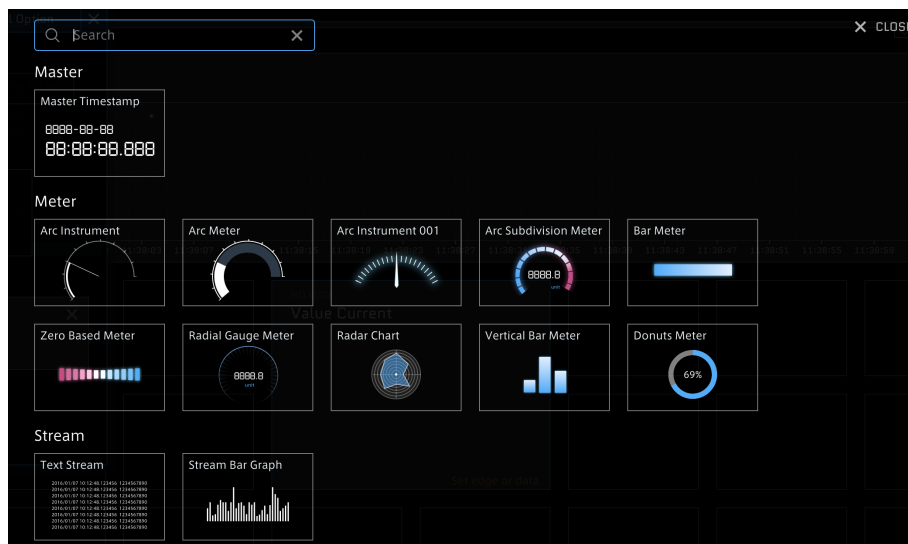


Fig. 28 Catalog of visual parts

The Line Graph is now in place on your dashboard.

3.3.4 Bind data to visual parts

Finally, select the data you want to visualize from the data list configured in [Create data settings](#) (p. 14) and bind the data to the visual part.

1. Right-click on the visual part to display the Panel Settings tab again. Click the button in Data Binding to open another window with a predefined list of data.

In this window, make settings starting from the top.

2. Click [Select Edge] and select `edge1`, because the CSV file will be uploaded as data from `edge1` in the later step.
3. Select the data you want to bind from the data list. In this tutorial, we will bind the X-axis acceleration. Select `sp_ACCX` from the data list.

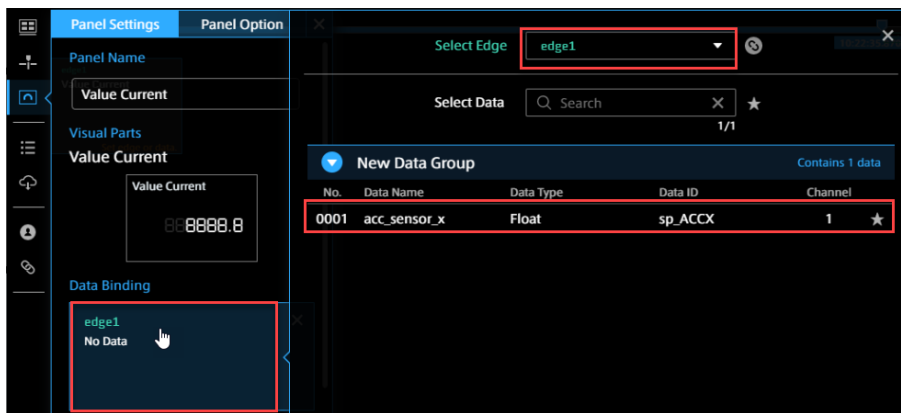


Fig. 29 [Panel Settings] window and data binding window

3.3.5 Bind multiple data to a visual part

Some visual parts allow you to bind multiple data. Since "Line Graph" can have multiple data, add acceleration values for Y and Z axes.

If you use a visual part that can bind multiple data, you will see [Add New Data] button in Data Binding. Click this and bind the Y-axis and Z-axis acceleration data in the same way as in the previous section.

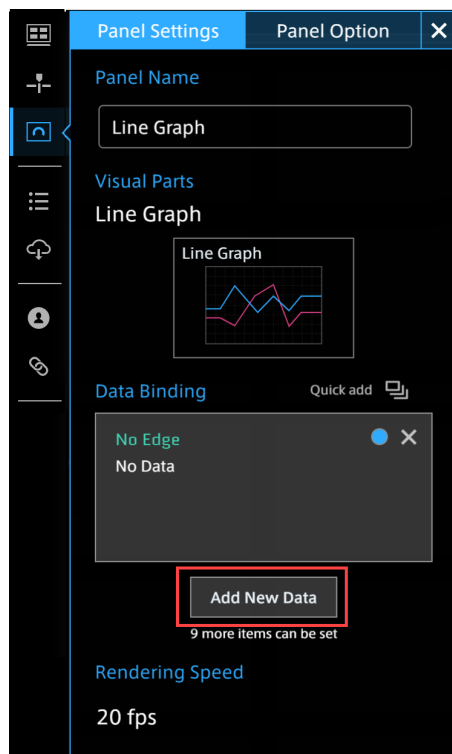


Fig. 30 [Add New Data] button

3.3.6 Change the position and size of visual parts

So far, we have set the visual parts in 2x2 size. Since the horizontal axis of a Line Graph is the time axis, it is easier to see the data in long horizontal panel. Adjust the position and size of the visual parts.

To move a visual part, drag and drop the visual part.

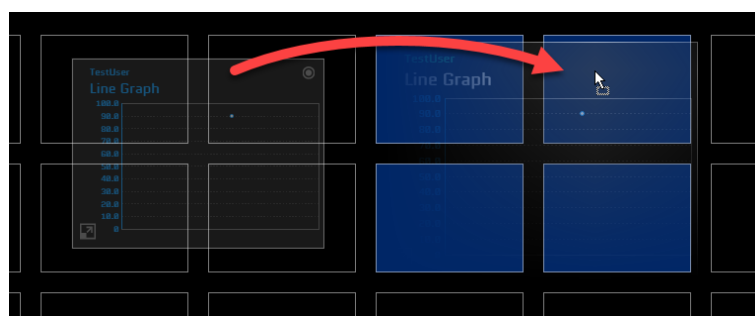


Fig. 31 Drag and drop a visual part to move

To resize a visual part, drag and drop a small circle that appears on the frame when you focus on the visual part.



Fig. 32 Resize visual parts

With these operations, you can arrange the visual parts in any size and shape.

3.4 Upload CSV file

Upload the CSV file and store the data in intdash. Upload the CSV file in the same way as described in [Upload a CSV file](#) (p. 8).

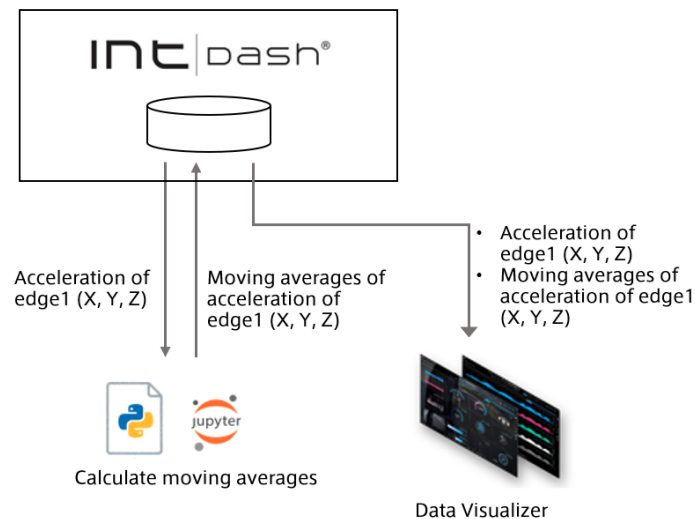
If you have already executed [Tutorial 1A: Visualize the sample data](#) (p. 5), you can proceed to the next step as the data exists on the intdash server.

3.5 Visualize uploaded data on dashboard

Visualize the uploaded data using the created dashboard. This procedure is also the same as [Visualize uploaded data on dashboard](#) (p. 10) in the previous tutorial.

04 Tutorial 1C: Process server data using intdash SDK

In the third tutorial, you will use the intdash SDK to access the data stored on the intdash server for simple processing. Also, the processed data is stored in the intdash server again, and the data before and after processing is compared on Data Visualizer.



4.1 Change dashboards

4.1.1 Register data settings for the processed data

Register appropriate data settings for the processed data.

1. Download the data settings file from the link below. (If you are viewing this PDF in your web browser, right-click on the following link and click [Save Link As].)

[Converted_Sensor.dat](#)

2. Register the parsing definition file in the same procedure as [Import a data settings file](#) (p. 6).

4.1.2 Load the screen for the processed data

Change the dashboard to display both data before and after processing.

1. Download the screen definition file from the link below. (If you are viewing this PDF in your web browser, right-click on the following link and click [Save Link As]).

[converted_sensor_tutorial1.scrn](#)

2. Register the Screen file in the same procedure as [Load a dashboard configuration file](#) (p. 7). Specify `edge1` as the source for all data in all panels.

4.2 Upload the CSV file

Store the CSV data you want to visualize in intdash. The procedure for uploading a CSV file is the same as [Upload a CSV file](#) (p. 8) in the previous tutorial.

If you have already executed [Tutorial 1A: Visualize the sample data](#) (p. 5), you can proceed to the next step as the data exists on the intdash server.

4.3 Retrieve and process data

Retrieve data stored on the intdash server using the intdash SDK, process it, and then store the results again on the intdash server.

Run the tutorial Jupyter Notebook [1_process-acquisition-data](#).

4.4 Display processed data

Visualize both the data before processing and after processing stored in the intdash server using Data Visualizer. This procedure is the same as [Visualize uploaded data on dashboard](#) (p. 10).

05 Tutorial 1D: Upload a large number of CSV files using intdash SDK

In the tutorials so far, we used a web browser to open the upload page and uploaded one file at a time. However, if you have a large number of CSV files, manually uploading them one by one will take a lot of time and effort. In Tutorial 1D, assuming such a case, we will create an upload script using the intdash SDK to automate the upload.

There is a [Jupyter Notebook](#) available to perform this procedure.

Note: The script introduced here is for CSV files only, but by using the intdash SDK, you can upload data in various formats such as JSON and images to intdash. For more information, see [intdash SDK for Python Samples](#).