SIEMENS

Insights Hub

Dashboard Designer

System Manual

03/2024

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

> ⚠ **DANGER**
>
> indicates that death or severe personal injury **will** result if proper precautions are not taken.

> ⚠ **WARNING**
>
> indicates that death or severe personal injury **may** result if proper precautions are not taken.

> ⚠ **CAUTION**
>
> indicates that minor personal injury can result if proper precautions are not taken.

> **NOTICE**
>
> indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

> ⚠ **WARNING**
>
> Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Table of contents

Introduction to Dashboard Designer 1

1.1 Introduction

Welcome

Dashboard Designer Add On dashboards offer an extensive library of visualizations that provide at-a-glance views of data for tracking metrics and drilling down into data. Select how you want to visualize your data and take advantage of our powerful query functions to precisely target the data to include and how to present it. In addition to being able to import dashboards, you can link your dashboard to internal and external Internet locations, as well as to other Dashboard Designer Add On dashboards.

Anyone can use Dashboard Designer Add On to create beautiful, informative dashboards. No programming knowledge needed.

Gain insight into your data by visualizing it in the way that makes sense to you.

> ✏️
>
> Dashboard Designer Add On is not available for Private Cloud.

Navigating to Dashboard Designer Add On

Access the application by clicking on the Dashboard Designer Add On icon on the Insights Hub Monitor Launchpad:

Visualizations and Plugins

These visualizations and plugins are included in your Dashboard Designer Add On subscription:

Basic

- Clock

- Pie Chart

- Table

- Gauge

- Text

- Dashboard List

- Graph

Advanced

- Bar Gauge

- Heatmap

- Discreet Panel

- Alert List

- Singlestat

Third-Party

- ImageIT

- Breadcrumb

- Cal - Heatmap

- Statusmap

- Traffic Light

- SVG, which faciltates sophisticated visualizations using JavaScript

> ℹ️
>
> For **guage**, **bar guage**, and **Singlestat** visualizations: there is an issue with color selection: you must double-click the color in the dialog box; a single click results in the box closing without saving your color selection.

Security Information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks. To protect against cyber security threats, Siemens's products and solutions implement and continuously maintain a holistic and state-of-the-art industrial security concept.
Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if, and to the extent such a connection is necessary, and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place. For additional information on industrial security measures you may want to implement, please visit the Industrial Security page on the Siemens website.

General Data Protection Regulations

Siemens observes the principles of data protection, in particular the principle of data minimization (privacy by design).
Dashboard Designer Add On processes and/or stores only the personal data necessary and essential for logging into the site, billing, internal user administration, and identifying authorized users. This data includes user:

- email addresses

- roles

- dashboard templates

- dashboard usage data

Users cannot store data anonymously, nor can data be pseudonymized, due to the essential need to identify authorized users.
Dashboard Designer Add On does not offer automatic deletion of data; data must be deleted manually. Please contact customer support about deleting data.

Plugins

Users cannot add 3rd party plugins to Dashboard Designer Add On. More plugins are coming in future releases. Please see the Visualizations and Plugins topic in this Help for more information on capabilities.

User Roles

Admin users can define roles, access, view, and edit permissions for their organizations' users, user groups, and teams. Access these functions by opening the Settings app on the Launchpad. This briefly describes the user roles in Dashboard Designer Ad On:

- Admin: has full access to all dashboard and tenant settings functionality Creator: can create and edit dashboards

- Viewer: can view dashboards

Usage Quotas

Dashboard Designer Add On subscriptions include a quota on the number of dashboards created. Please see Technical Limitations.

Identity and Access Management 2

2.1 Identity and Access Management for Dashboard Designer Add On

For Organizations

Your subtenants are represented as "Organizations" (Orgs) in Dashboard Designer Add On. Orgs are automatically created when a subtenant user first logs in.

Default User Roles

Dashboard Designer Add On users can access dashboards and assets according to the user role assigned to them:

- Admin: allows full access to all Dashboard Designer Add On functionality. Admin users can define and override dashboard editing and viewing settings for individual users and groups, allowing control of user access. Only users with the Admin role can access and switch between Orgs (subtenants).

- Creator: allows creating and editing dashboards in Dashboard Designer Add On.

- Viewer: allows users to view dashboards in Dashboard Designer Add On, but viewer users cannot make changes to dashboards.

Teams

Teams are groups of users.

Navigating
3

3.1 Navigating Dashboard Designer Add On

View Modes

Dashboard Designer Add On has three view modes that you can switch between by clicking the Cycle view mode icon in the upper right corner of the screen. Here is a brief description of the view modes:

Home when you open Dashboard Designer Add On, your home screen displays the navigation bar, starred and recently-viewed dashboards, and a link to the Monitor Legacy Dashboard app.

Basic when you open a dashboard, it displays on the basic screen, along with the navigation bar and dashboard action icons.

Kiosk mode displays when you click the Cycle view mode icon once. It hides the navigation bar and dashboard action icons, and expands the display to full screen. Press the Esc key to see the navigation bar and dashboard action icons.

Side Panel Icons and Navigation

This image shows the navigation and submenu items for each:
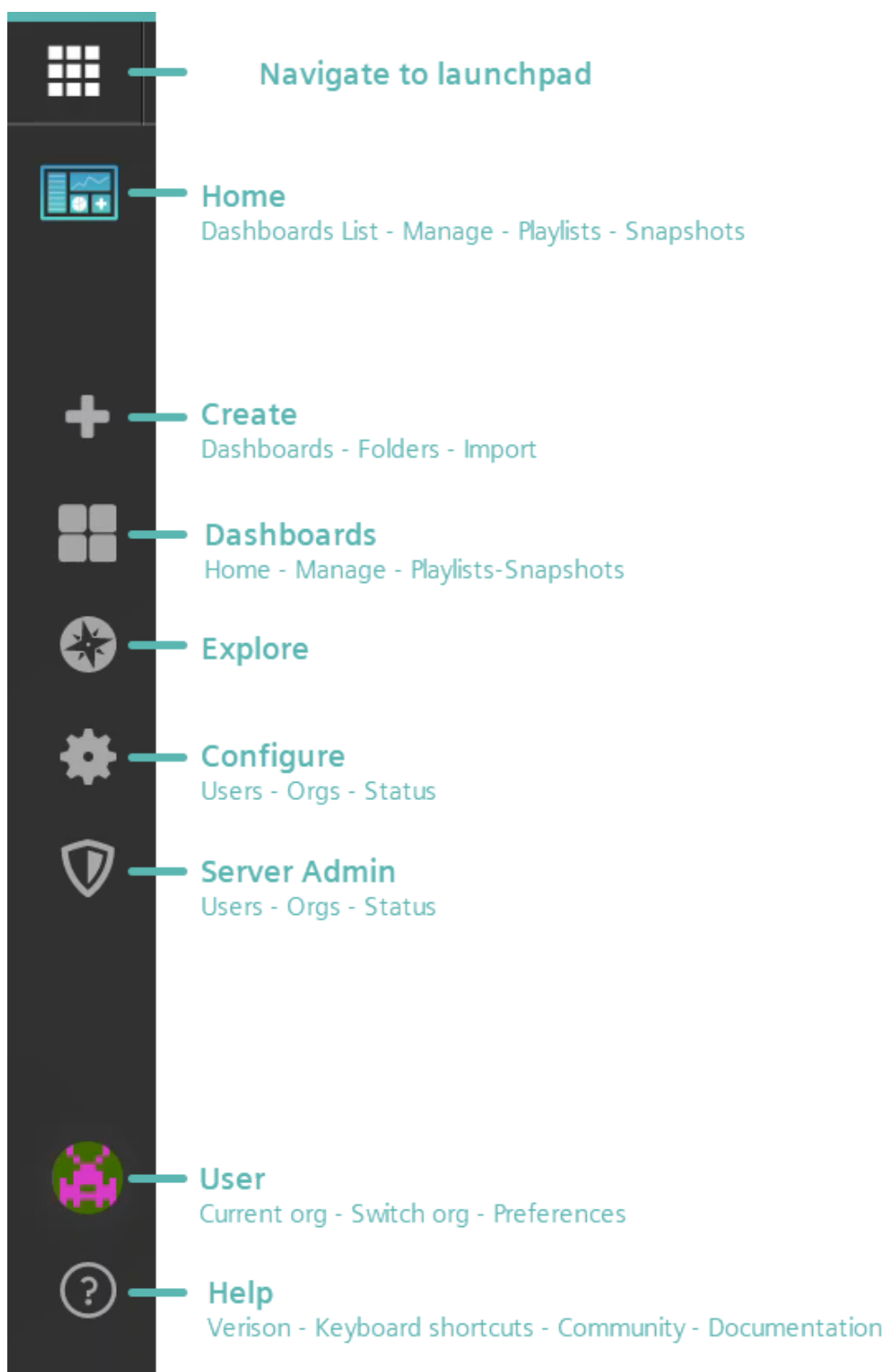
!!! note The last two icons, "Configure" and "Server Admin" both navigate to the Server Admin page and display the same content.

Plugins

Please refer to the "Visualizations & Plugins" topic in this Help documentation for more information.

User profiles

You can open your user profile settings by clicking the icon. Your icon will differ from the example shown in this image:
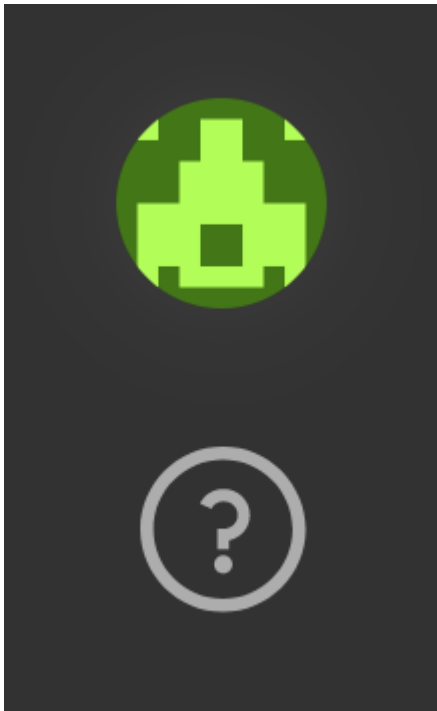
Keyboard Shortcuts

This image shows keyboard shortcuts:

Getting Started 4

4.1 Getting Started

Prerequisites

Before starting, please make sure you have:

- Editor/admin access to Dashboard Designer Add On

- Some asset data in your current set-up

- Dashboard Designer Add On supports Chrome and Firefox Internet browsers.

- Internet Explorer is not supported.

- Dashboard Designer Add On is not available for Private Cloud.

- Integrated Data Lake (IDL) data sources do not support annotations.

Best Way to Learn

Open another page of Dashboard Designer Add On side-by-side with this tutorial so you can try out some steps.
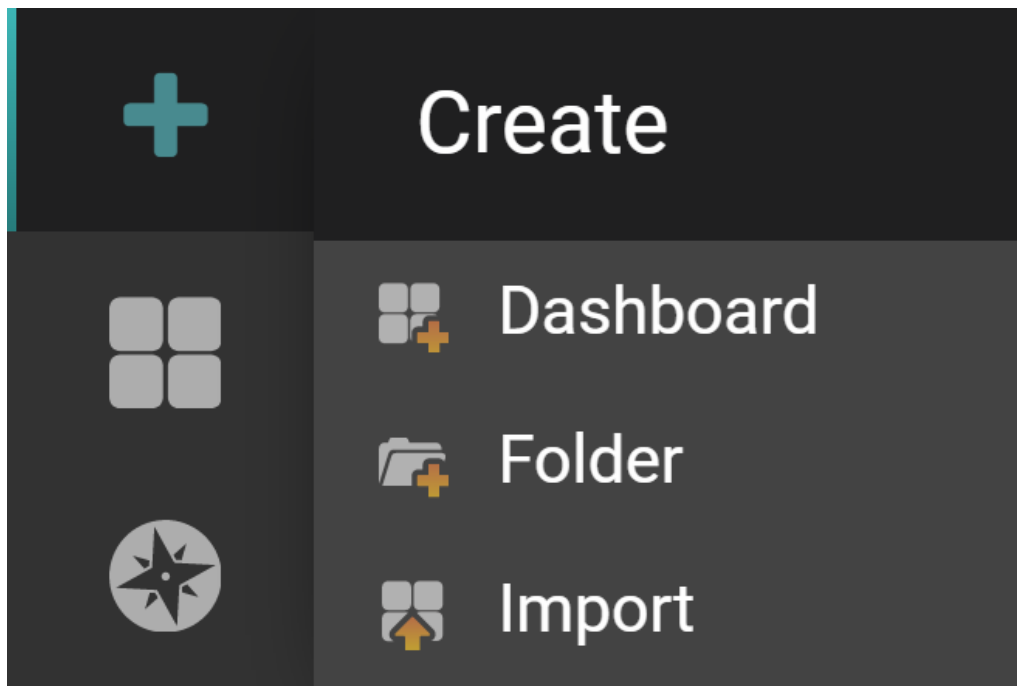
Creating Dashboards

A dashboard gives you an at-a-glance view of your data and lets you track metrics through different visualizations.

Panels

Each panel on a dashboard represents a part of the story you want your dashboard to tell. A panel consists of a:

- Query: defines which data to display

- Visualization: defines how the data displays

The Panel Edit Page

The Panel Edit page displays three tabs on the left:

- Query: where you create the query to pull your data in Dashboard Designer Add On. created the data query to this panel earlier in step 3.

- Visualization: where you can select a plugin for visualizing the data. You can also change the visualization settings of the plugin.

- General: displays the panel's title and settings, such as if the panel hyperlinks to another dashboard.

How to Create a New Dashboard

Follow these steps to create new dashboard:

Hover over the + button on the side panel, and click "Dashboard" to create a blank
1. dashboard with an empty panel.

2. Click "Add Query" to open the panel edit page.
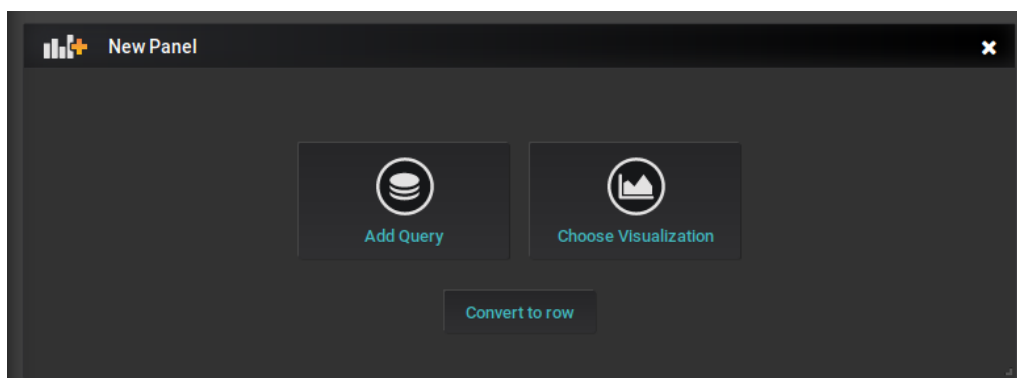
3. If you have asset data available, you can continue this tutorial with your data; if not, you can use the provided test data.

Getting Your Data into Dashboard Designer Add On

Follow these steps to retrieve your data:

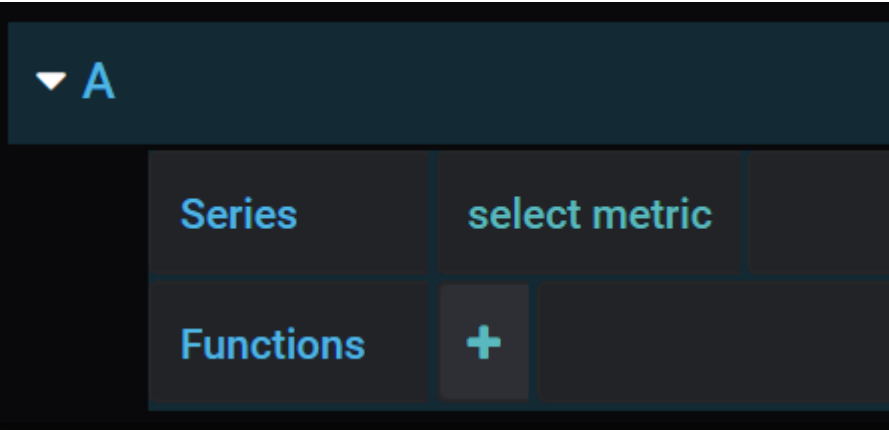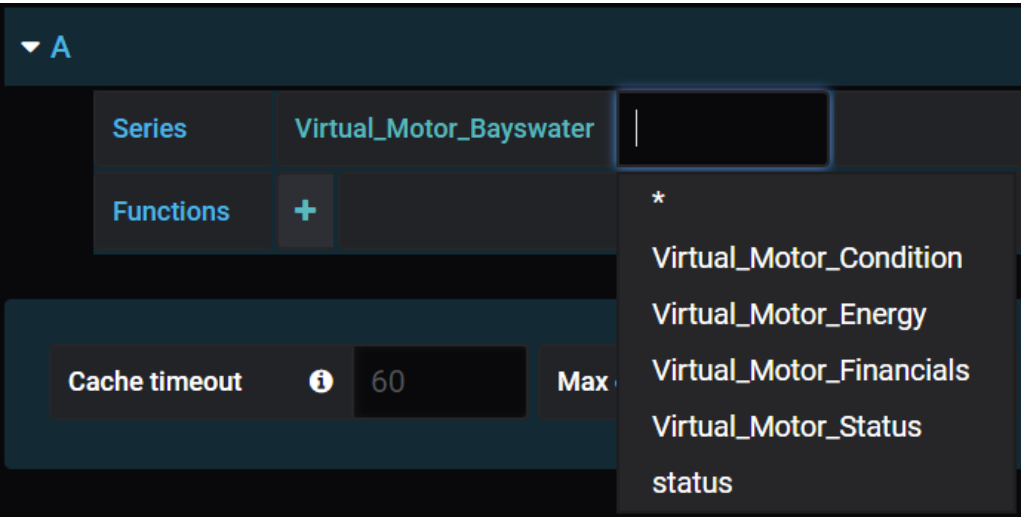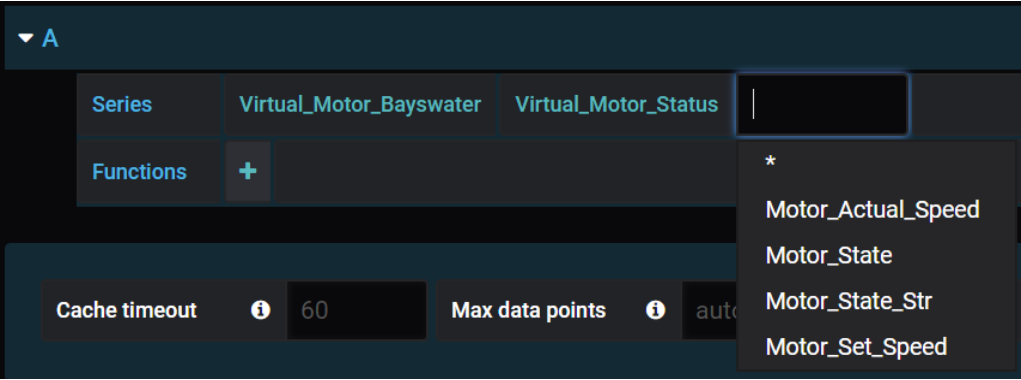1. Click "select metric" or enter an asset name in the search field. Aspects included in the asset display.

2. Select an aspect. The aspect's variables display.

3. Select a variable.

> ℹ
>
> If no data is available, it's likely that the default time range does not contain data from your asset. Try selecting different time ranges from the time selector drop-down list. If you continue to experience issues in retrieving data, try using the generated test data for the next steps.

> ✏
>
> Dashboard Designer Add On does not currently support variables assigned directly to asset types.

Connecting to Data

If you are already getting data, skip step 1 below; otherwise, follow these steps to connect data:

1. Select testData from the drop-down list. Random data is used by default, but you can use other data scenarios.

2. When your data shows, click the Esc key (or back button) to return to the dashboard page. Your dashboard displays with the new panel.

3. To create another panel, click the "Add panel" button.

Adding a Visualization and Plugin

Follow these steps to add a visualization:

1. Click the "Choose Visualization" option. The Visualization tab opens under the "Panel edit" page.

2. Click the graph drop-down list and select "Gauge" from the list.

3. Click the "Query" tab.

4. Select some of your data or test data. When the data comes in, press the Esc key or the Back button to go back to the Dashboard page.

5. You can duplicate the panel using one of the methods discussed in the "Additional Panel Actions" topic, which appears right after the Visualization image below.

Your dashboard should now look similar to this:
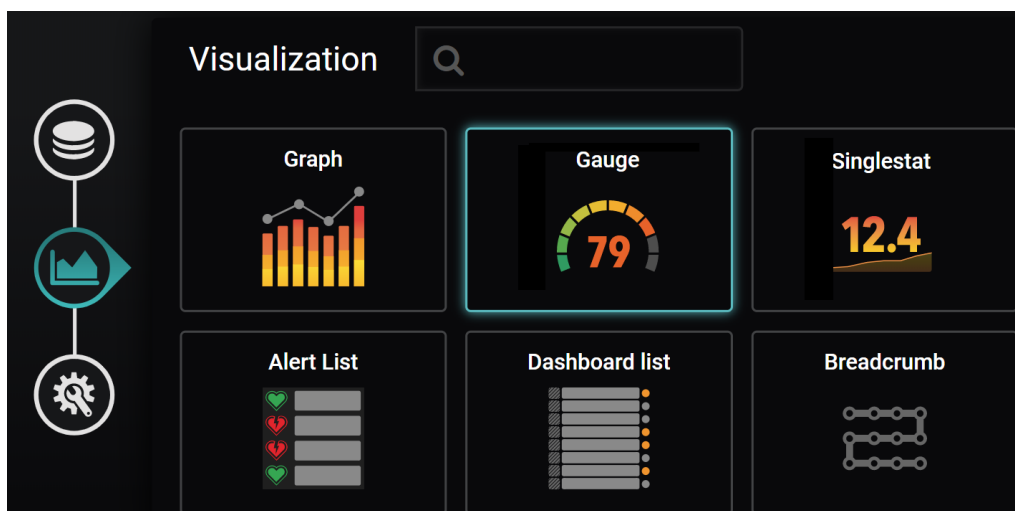
Additional Panel Actions

Some additional actions you can take with panels are:

- Resize panel: click and drag the lower right corner of the panel to resize it up or down.

- Duplicate: select "More" from the drop-down list at the top of the panel, then click "Duplicate". Alternatively, you can left-click anywhere on the panel, then press "p" and "d".

After duplicating the panel, we recommend trying a different query so different data displays.

Saving Your Panel(s)

Follow these steps to save your panel:

1. Click anywhere on the panel and press "e" on your keyboard, or click the heading of the panel and select "Edit". The "panel edit" page displays.

2. Navigate to the "General" tab, and enter a name for the panel. Repeat as needed for other panels in your dashboard.

Saving Your Dashboard

Follow these steps to save your dashboard:

1. Click the 'save disc' icon in the top right corner of the dashboard. The save dialog displays.

2. Enter a name for the dashboard.

3. Select (optional) a folder to store the dashboard.

Creating Dashboards 5

5.1 Creating Dashboards

This topic guides you through the aspects of creating a dashboard.

> ✏️
>
> Only Admin and Creator users can create dashboards.

About Dashboards

Dashboard Designer Add On dashboards offer an extensive library of visualizations that provide at-a-glance views of data for tracking metrics and drilling down into data. Once you find the right visualization for your data, you can take advantage of the powerful query functions to specify the exact data to include, and how to present it. In addition to being able to import dashboards, you can link your dashboard to internal and external Internet locations, as well as to other Dashboard Designer Add On dashboards.

Dashboards are composed of individual panels arranged on a grid, and each panel represents a part of the story you want to tell through your dashboard.

Each panel consists of a query and a visualization. Queries define the data you want to display, and visualizations define how it's displayed.

In addition to the panel, query, and visualization, Dashboard Designer Add On includes plugins--tools you can use in conjunction with visualizations for even more ways to display data on a panel.

Dashboard Data Limitations

Dashboard Designer Add On supports up to ten columns and three thousand (3,000) rows of data in a dashboard.

Panel Edit Page Tabs

The "Panel edit" page contains three tabs on the left side. The tabs are:

- **Query**: where you create queries that bring data into your panel.

- **Visualization**: where you select widgets and plugins to visualize data according to your preferences.

- **General**: where you select panel titles, links to other panels or dashboards and other general settings.

How to Create a New Dashboard

Follow these steps to create a new dashboard:

1. Hover over the **+** button on the side panel, and click "Dashboard". A new dashboard with a blank panel displays.

2. Click "Add Query" to open the panel edit page.

3. If you have asset data available, you can continue to the next steps using this data; if not, you can use the provided test data.

Getting Data into a Dashboard

The sections below guide you in using differing kinds of data. You can proceed using data from:

- Internet of Things (IoT)

- Integrated Data Lake (IDL)

- Test data

See the section below that applies to the data you are using.

Using IoT Data

Follow these steps to read your data from its IoT location and write it to your dashboard:

1. Click "Select metric" to select the asset that has the data you want to read.

2. Use search to locate an asset or enter its name in the asset field and select the "asset". Its aspects display.

3. Select an "aspect" in the next select metric box. Its variables display.

4. Select all, or select the variables to include.

5. Click "Add Data Source". The path to your data source displays in the Data Source Location section.

If no data is available, sometimes it's because the default time range does not contain asset data. To remedy this, try the following:
Click the time picker drop-down and try various time ranges. If you still have issues getting data, try the next steps using generated test data.

> ✏️
>
> Assigning a variable directly to an asset type is not yet supported in Dashboard Designer Add On.

Using IDL Data

Follow these steps to read your data from its IDL location and write it to your dashboard:

1. Hover over the **+** button on the side panel, and click "Dashboard". A new dashboard with a blank panel displays.

2. Click "Add Query". The Panel edit page displays.

3. Select the Integrated Data Lake radio button. The IDL pop-up window displays.

4. Select the file type of your data (CSV or Parquet). Folders display according to the file type you select.

5. Select the folder that contains your data or use the search bar to locate it. Folder contents display to the right.

6. Select check boxes for each field to include in the query.

7. Click "Add Data Source". The path to your data source displays in the Data Source Location section.

Using Generated Test Data

Is you already read data from your data source locations, skip these four steps that generates data.
Follow these steps to use generated test data:

1. Select testData from the drop-down.

2. By default, 'random walk' (random data) is used. Feel free to try other data scenarios.

3. Once data shows, click the Esc key or Back button. Your dashboard with the new panel displays.

4. Continue to add panels by clicking the "Add panel" button.

Specifying Visualization Options

Follow these steps to specify visualization options for a panel, beginning on the Panel edit page:

1. Click the "Graph" drop-down and select "Singlestat" from the list of available plugins.

2. Click the "Query" tab.

3. Select the name of the data you want to use from the drop-down list.

4. Once data shows, click the Esc key or Back button. Your dashboard with the new panel displays.

5. Continue to add panels by clicking the "Add panel" button.

6. Resize the Singlestat panel by clicking and dragging its lower right corner.

7. Duplicate the panel by selecting "More" from the drop-down at the top of the panel, then select "More", and "Duplicate".

We recommend giving the duplicate panel a different query to bring in different data.

How to Save your Dashboard

Follow these steps to save your dashboard:

1. Navigate to the "Panel edit" page and select the "General" tab.

2. Enter a name for the panel in the "Name" field.

3. Repeat for any unnamed panels on your dashboard.

4. Click the "Save" icon in the top right corner of the dashboard.

5. Enter a name for your dashboard and select a folder where you want to store it (optional).

Using Integrated Data Lake Data Sources 6

6.1 Using Integrated Data Lake (IDL) Data Sources in Dashboard Designer Add On

IDL data sources are accessible in the data sources drop-down list when you create a new panel and select a visualization, or when you create or edit a query. This allows you to:

- Bring CSV and Parquet data in from IDL

- Visualize multiple IDL queries in one widget

- Visualize IoT and IDL data in one widget

IDL Data Source Limitations

- Annotations are not supported for IDL data sources.

- A maximum of 4,000 records is supported in IDL queries.

- Aggregated data is not currently supported in queries, but is planned for a future release. The exception is the Gague widget, which auto-aggregates the data returned by a query.

- A query time range of up to five years is supported.

- Access controls may not extend to all data in IDL files and folders. For example, if two IDL files with access restrictions reside within an IDL folder that is unrestricted, and a user queries the unrestricted folder, all the files are returned, irrespective of restrictions on the IDL files or the user who creates the query.

- IDL data sources do not support annotations.

Example of the IDL Data Source Selection
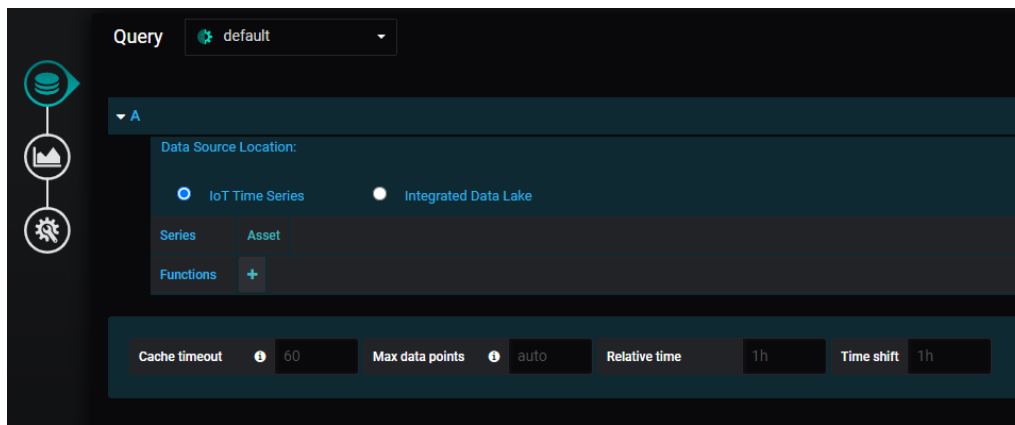
How to Write IDL Data to Your Dashboard

Follow these steps to read your data from its IDL location and write it to your dashboard:

1. Hover over the **+** button on the side panel, and click "Dashboard". A blank dashboard with an empty panel displays.

2. Click "Add Query". The Panel edit page displays.

3. Select the Integrated Data Lake radio button. The IDL pop-up window displays.

4. Select your data's file type (CSV or Parquet).

5. Click the folder containing your data or use the search bar to locate it. Folder contents display to the right.

6. Select check boxes for each field to include in the query.

7. Click "Add Data Source". The path to your data source displays in the Data Source Location section.

Here is an image of the Data Source Loation section:
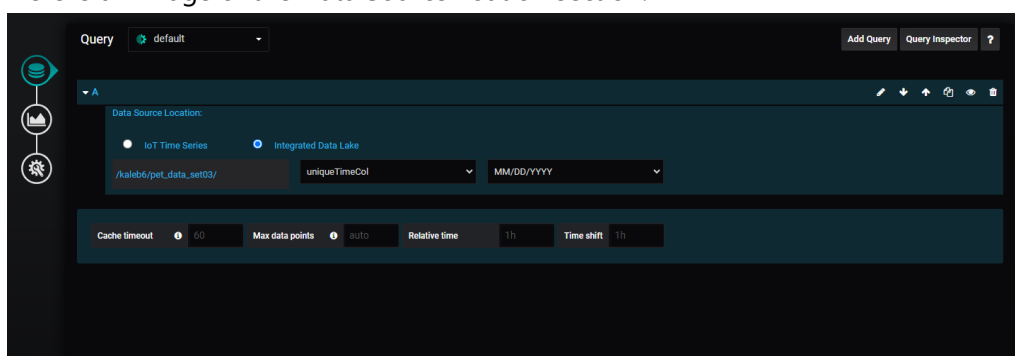
How to Select the Date Options for an IDL Data Source

Follow these steps to format the date fields:

1. Select a value from the time column drop-down list.

2. Select a date-time format. Data renders at the top of the screen according to your entries.

3. Continue to add additional queries by clicking the Add Query button and repeating the steps above.

Tips and Tricks in Dashboard Designer 7

7.1 Welcome to Tips and Tricks for Using Dashboard Designer Add On

While easy to use, Dashboard Designer Add On has many techniques for creating more advanced visualizations and user interactions. This section's topics show some of these advanced techniques.

Since Dashboard Designer Add On is built using a variety of commonly-used open-source applications, you can find many answers on the Internet by searching with the keyword "grafana".

Optimizing Dashboard Performance

When using data sources, dashboard performance usually depends on how the data is structured and how you are retrieving it. This section presents many tips for optimizing your dashboard's performance.

It is important to plan ahead during the design phase to maximize the performance of your dashboards. Considering the points below will help ensure the best performance.

For IoT data sources, ensure the aspect's variables have similar sampling rates

Let's consider, for example, an asset type with this aspect:

- Aspect 1 Sampling rate

- Batch_ID 1 Hour

- Material_ID 1 Hour

- Energy_Usage 15 Seconds

- Motor_Speed 15 Seconds

If you try to retrieve the raw timeseries data of 'Batch_ID' in the past 24 hours, the query request will have to filter through and ignore all the variables that have a 15 seconds sampling rate in the past 24 hours--resulting in a slower response due to the nature of the timeseries API.

Best practice: group variables with similar sampling rates to make the API requests more efficient. For example:

- Aspect 1 Sampling rate

- Batch_ID 1 Hour

- Material_ID 1 Hour

- Aspect 2 Sampling rate

- Energy_Usage 15 Seconds

- Motor_Speed 15 Seconds

Use the correct functions

By default, all data requests are aggregated. If your visualisation only shows 1 data point (e.g. singlestat, gauge, bar gauge), consider using this function: 'MSmostRecentValueWithinTimeRange'.
For example:

- 'MSmostRecentValueWithinTimeRange' runs faster than 'MSmostRecentValueWithin90Days'

- 'MSrawTimeseries' function can have a payload of maximum 4000 data points; if your visualization uses less data points, you should set a limit via the Max Data Points setting.

Consider a longer auto-refresh time setting

A short auto-refresh cycle slows dashboard performance. Also note that a refresh will also run when the response time is longer than the auto-refresh cycle.

Events as annotations

Use annotations to integrate event data into your graphs. Annotations are visualized as vertical lines and icons on all graph panels. Hover over an annotation icon to see the description & severity for an event.
To add events to your dashboard, navigate to:
Dashboard Settings > Annotations > Add Annotation Query and follow these steps:

1. Select a repsository from the Data Source drop-down list.

2. Enter an asset name to retrieve its events. Optionally filter for events of a particular severity by adding a severity field after the asset name.

Using variables to create a template dashboard

Variables allow you to create more interactive and dynamic dashboards by replacing hard-coded elements such as assets and aspects in your metric queries with variables.
A variable is a placeholder for a value that you can use in:

- metric queries

- panel titles

If you have multiple assets that share the same asset type, it is useful to create a reusable dashboard. This makes it possible to quickly browse through all of the assets/aspects by using variables.

When you change a value by selecting from the drop-down list at the top of the dashboard, the panel's metric queries change accordingly.

How to add an asset variable

To add an asset variable, navigate to Dashboard Settings > Variables > Add Variables and follow these steps:

1. Enter a name for the variable (e.g. 'Asset'), followed by the repository name in the Query Options field under Data Sources.

2. Enter the * symbol in the Query field in Query Options.

3. Add a regex filter to the query options to refine the filter. For example, if you have multiple pump assets named 'pump_1', 'pump_2', 'pump_3' ... You can filter for all the pump assets by entering "pump.*". This filters for all asset names beginning with "pump". View the preview of values to be sure your variable are correct.

4. Optionally check the potential values for your variable are correct under "Preview of values".

5. When finished, click "Add" on the bottom of the screen to add the variable. The variable displays in your dashboard "Asset" drop-down list.

6. To use the variable enter '$variableName' when you create a new data query--in this example, '$Asset'. Enter the aspect and variable of the query as you normally do.

When you finish your query, you can change the assets on the panel.

How to add an aspect variable

To add an aspect variable, navigate to Dashboard Settings > Variables > Add Variables (if you already created a variable, select "New") and follow these steps:

1. Enter a name for the variable (e.g. 'Aspect').

2. Enter assetName.* in the query. This queries for all the aspects of the asset you specified.

3. Add a regex filter to the query options to refine the filter for the aspect's variables.

4. Optionally check that the values for the variable are correct under "Preview of values".

5. When finished, click "Add" on the bottom of the screen to add the variable. When you have created the variable, you should now see the 'Asset' dropdown on your dashboard, allowing you to change between the assets.

How to use the aspect variable

To use the variable, enter '$variableName' when you create a new query; in this example, '$Asset' in the asset metric and '$Aspect' in the aspect metric. Enter the variable of the query as you normally do.

Once you set your query, you can make changes to the aspects on the panel.

User interactions with drill-downs and dashboard links

When creating dashboards, it is important to consider your users and that they have different needs.

Some users want their dashboard to **quickly answer questions** like, '*Is there an emergency alarm?*, or '*Has my machine stopped?*, or '*How are today's KPIs tracking?*

Some users want to **drill down to find any issues or trends**.

Some users want **both** quick Q&A and drill-downs on the same dashboard, however, too many questions answered on a single dashboard makes it messy and complex.

Best practice: create multiple dashboards and allow users to navigate between them.

Understanding dashboard URLs

Every dashboard in Dashboard Designer Add On has a unique UID and unique URL. Look at the current URL of this tutorial page; it can be broken down into sections:

baseURL /Dashboard Designer Add On /d /{dashboardUID} /{dashboardName}

For example:

tenant-Dashboard Designer Add On-silopsms.eu1.mindsphere.op/Dashboard Designer Add On/d/TzRQFzkGz/factory-dashboard

Depending on your dashboard, the URL can have other parameters such as date time and variables:

... ?orgid ={organisation(subtenant) ID} &from ={timeFrom} &to ={timeTo} &var- {variableName} ={variable}

For example:

tenant-Dashboard Designer Add On-silopsms.eu1.mindsphere.op/Dashboard Designer Add On/d/G2JVBTwEk/factory-dashboard?orgID=1&from=1586318878357&to=now&var-Asset=Motor_A

How to create a drilldown link on a dashboard panel to another dashboard

> ✏️
>
> Certain plugins may not support drilldown links.

Navigate to the Panel Edit page for any dashboard and go to the General tab. Follow these steps to create a drill down link to another Dashboard Designer Add On dashboard:

1. Select Absolute from the Type drop-down list in the Drilldown Links section.

2. Enter the dashboard URL in the URL field beneath the Type field. The base URL is already included, so subtenant (organization) and dashboard names are not required.

3. Change or skip the 'time' and 'variable' fields depending on whether you want to carry over your current dashboard settings.

4. To customize time and variable settings, turn off Include time range > Include variables, and manually enter them in the URL params field; for example:
&var-Asset=Pump_05&from=now-90d&to=now or include them in the absolute URL.

> ℹ️
>
> You can use variables in the URL and Url params, e.g, &var-Asset=$Asset.

Creating dashboard navigation using dashboard links

Dashboard links appear in the top right corner of your dashboard and act as links to either dashboards or external links. You can add tags to dashboards under Dashboard Settings > General > Tags.

How to create a dashboard link

Follow these steps to create a dashbord link:

1. Navigate to Dashboard Settings > Links > Add Dashboard Link.

2. Select links from the Type drop-down list.

3. Select the dashboard you want to link to. 4. To link to all dashboards, select dashboards from the Type drop-down list.

Creating a "batch list" or "shift list" with tables

Selecting a link on a dashboard allows you to navigate from your dashboard to any link, internal, another dashboard with a specific URL, or external, for example a customer support website.

> ✏️
>
> We recommend reading section 4 which covers user interactions with drill downs and dashboard links before starting.

Even though Dashboard Designer Add On has a comprehensive time picker, industrial use cases often require that dashboard data be segregated by *shifts* or *batches*. You can achieve this with a table panel. Depending on the available raw data, there are multiple ways to create the table. Here are some examples.

Best practice: data structure with start and end time available in "Unix timestamp" format. If possible, record the batch or shift start and end time in Unix format, this allows the datapoint to

be directly used in Dashboard Designer Add On's URLs without any transformation. Here is an example of this data structure:

Variable Data Type Example Data Explanation

Batch_ID STRING/INT/DOUBLE B1001 A batch or shift number Start_Time STRING/INT 1577845800000 2020-01-01 13:30:00 in Unix format (UTC) End_Time STRING/INT 1577856600000 2020-01-01 16:30:00 in Unix format (UTC)

> **ℹ**
>
> You can use the Visual Flow Creator (VFC) app to convert standard time to Unix epoch. If you have start and end events (e.g. in boolean format) you can also use VFC to create the start and end times based on the events.

By default, UTC time is converted to your local time. If your time is not stored in UTC, use can also use math functions to convert it.

How to create the batch table

Follow these steps to create a batch table:

1. Create a panel using the Table plugin.

    Query for the required data points. Strings can only be retrieved using the MSrawTimeseries function. This example queries the entire 'batch_info' aspect via a * wild card. This applies the
2. alias function to our results.

3. Navigate to Column Style > Add column style in the Visualization tab of the table panel.

4. Enter the name of the column for your batch/shift data, and enable 'Render value as link'.

5. Enter the target dashboard URL in this format:
/Dashboard Designer Add On /d /{dashboardUID} &from ={timeFrom} &to ={timeTo}

Formatting the URL to connect to a specific time period

Use a dynamic {timeFrom} and {timeTo} in the link URL by using values from the 'Start Time' and 'End Time' columns.

You can use special variables to specify cell values in the URL --- ${__cell} refers to current cell value. ${__cell_n} refers to Nth column value in the current row. Column indexe start at 0. For example, ${__cell_1} refers to second column's value.

The variable '{timeFrom}' time is the 'Start Time' variable, from the 3rd column, and you can reference it using ${__cell_2} and the '{timeTo}'. Time is the 'End Time' variable, from the 2nd column. Reference it using ${__cell_1}.

> **ℹ**

You can hide unneeded columns in the bathch/shfit table by selecting Hidden from the Type drop-down. This preserves the column positions in the table so the URL variables will still work.

If you only have start and end events

You can retrieve the Unix timestamp of an event using the timeOf function.

If you do not have end times available

If you are only recording a batch/shift ID you can still use the timestamp of your batch/shift datapoint to segregate data. Variable Data Type Example Data Explanation Batch_ID STRING/INT/DOUBLE B1001 A batch or shift number

When a timestamp of timeseries data is retrieved by Dashboard Designer Add On, it is stored in Unix epoch format by default. The steps to create a batch/shift list from this is very similar. The only difference is when referencing the date & time value in your table URL, you will have to use {_cell_n:raw} to read the Unix epoch value of the timestamp. For example, {Dashboard Deisgner Add On/d/TzjGZFkEp?from=$(_cell_0:raw}

Show assets on a floorplan or map using the Imageit plugin

You can use the Imageit plugin to create a map or floorplan of your site and overlay it with data. The Imageit plugin can show "sensors" which contains a metric value and a position value. You can add a "sensor" to your image by clicking on Add Sensor under the visualisation tab, and:

- In the sensor settings, select your Metric values from a dropdown; this is populated by the your query results which can be renamed using the alias function.

- The X position of the "sensor" is a value between 0 and 100. 0 being the leftmost position on the image, 100 being the rightmost position on the image.

- This value can be the Y-position of the "sensor" is a value between 0 and 100. 0 being the topmost position on the image, 100 being the bottommost.

If your asset's location is static

You will have to manually type in the X & Y position of your "sensor" in the sensor options.

The Grafana plugin that allows the user to drag the sensor around is not supported in Dashboard Designer Add On.

If your asset is moving

You will need to first convert your asset's position data into an X & Y position format, then scale it to 100. You can use the math function for scaling: $X\_pos\_scaled = X\_pos / (X\_max - X\_min) * 100$ $Y\_pos\_scaled = Y\_pos / (Y\_max - Y\_min) * 100$

Defining edit/view rights for specific users or teams

Further user access control can be managed using the Permissions settings. You can:

• Overwrite the default Dashboard Designer Add On roles, delete the editor and viewer role by clicking on the red *.

• Give a specific user or team the rights to view, edit or administrate the dashboard, click Add Permission.

How to Import a Dashboard

To import a dashboard, follow these steps:

1. Go to JSON Mode. This page shows the raw code behind your current dashboard.

2. If you want to save a local copy of the dashboard, copy the JSON text and save it in a ".json" file.

3. On the sidebar, hover your mouse over the * icon and select Import.

4. Paste the copied JSON model and click Load.

5. If you have an identical dashboard in your current tenant and do not wish to overwrite it, give your dashboard a new name.

6. Click the change button and delete the existing UID so the system generates a Unique identifier (UID) for the dashboard.

7. Click Import.

Copy a panel

To copy a panel, click the top of the panel and select Copy from the More menu. and go to More > Copy. Then paste the panel by creating a new panel and select Paste copied panel.

Duplicate a panel

To duplicate a panel,select the panel and press p + d, or click on the top of the panel and go to More > Duplicate.

Copying a panel's JSON model

To copy a panel's JSON model, click on the top of a panel and go to More > Panel JSON. In the new window, click Copy to clipboard. On a new panel, click on the top of the panel and go to More > Panel JSON, then paste the copied panel in the new window and select Update.

Official IoT TS Aggregates Service documentation: Version 4.x - What's New?

10.1 IoT Aggregates v4 changes and limitations

- The v3 to v4 upgrade changes have impact to Msaggregation > MsautoAggregation > MsboolAggregation functions only. If no function is selected in the Query configuration, the MsautoAggregation function is applied by default.

- Changes to supported aggregation units:

- second – not supported in V4 (removed)

- minute – supported

- hour – supported

- day – supported

- week – supported (added)

- month – supported (added)

- inute – aggregation unit range limit: 48 hours. If selected date range exceeds 48 hours the user gets an error message.

- hour – aggregation unit renage limit: 30 days. If selected date range exceeds 30 days the user gets an error message.

- day, week, month – aggregate unit range limit: 5 years. If selected date range exceeds 5 years the user gets an error message.

v4 introduces API requests rate limit – the number of allowed requests per minute depends on your subscription.

Possible drawbacks and solutions

The cases described below can be observed only in conjunction with Msaggregation > MsautoAggregation > MsboolAggregation functions. If no function is selected in the Query configuration, then the function is applied by default.

Minute aggregation unit

If the selected date range exceeds 48 hours for the minute aggregation unit then the widget will display an error message:
The problem can be resolved by either of the options:

- Select less than 48 hours date range for the dashboard

- Use hour aggregation unit instead of minute.

- Override widgets date range using Relative time configuration

10.2.2 Hour aggregation unit

If the selected date range exceeds 30 days for the hour aggregation unit then the widget will display an error message:
The problem can be resolved by either of the options:

- Select less than 30 days date range for the dashboard Use day aggregation unit instead of hour Override widgets date range using Relative time configuration

10.2.3 API rate limit

Depending on the dashboard configuration the user may start getting API rate limit exceeded error messages. This can be caused by big amount of aggregation functions used by dashboard configuration or high dashboard refresh rate. If the API rate limit is exeeded the widget will display an error message:
The problem can be resolved by either of the options:

- Increase dashboard refresh interval

- Decrease amount of aggregation functions on the dashboard

- In certain cases (e.g. Gauge, Bar Gauge) it is advised to use MSmostRecentValueWithinTimeRange function instead of aggregation

10.2.4 How to get sub-minute aggregation data

As second aggregation unit is not support by v4 the sub-minute data can be visualized by:

- Use the MsrawTimeseries function to get raw time series data

- Use VFC application to define custom aggregation logic

Functions 8

8.1 Dashboard Designer Add On Functions

Functions in Dashboard Designer Add On are used to associate or transform raw data into useful formats for creating meaningful dashboards. Functions only display when you select Insights Hub or Default data sources.

Data requested by Dashboard Designer Add On is automatically aggregated by default. If your raw data must be queried in a different way, use the Functions feature to modify the query. Functions appear when you select a "Insights Hub" or "Default" data source. Functions you add to a query will override default Query settings.

> ✏️
>
> Dashboard Designer Add On is not available for Private Cloud.

Functions Overview

A query can have multiple functions and you can access them by opening a panel and navigating to the Query page:
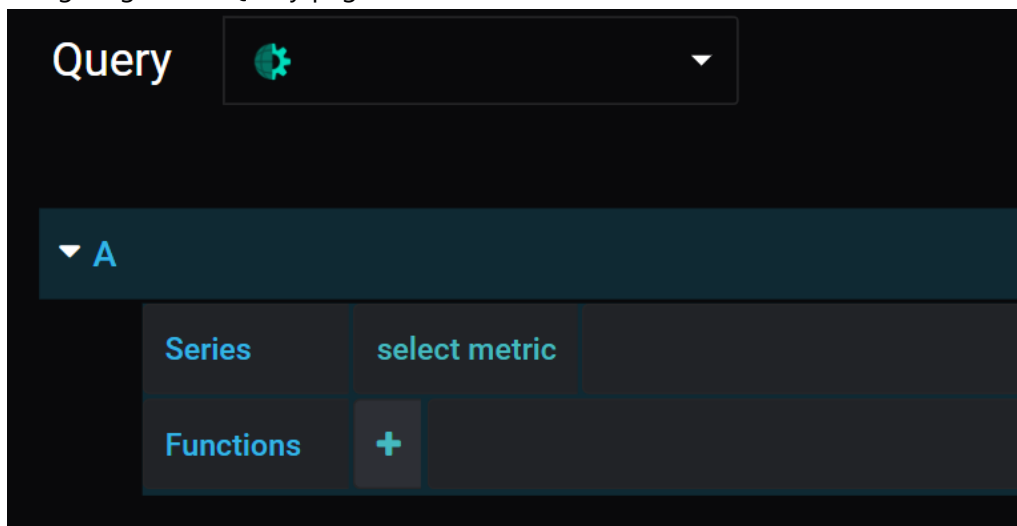
Adding a Function to a Query

When you build a query, you can add functions to it by hovering over the + symbol. Click the + symbol to open a scrollable drop-down list of functions. You can add multiple functions to a

single query.

How to Add Functions to a Query

Follow these steps to add a function to a query:

1. Click the + icon next to "Functions".

2. Hover over the data source name ("Insights Hub") and select a function from the drop-down list.

Removing a Function from a Query

Click a function name to open a pop-up window, and select "x" to remove a function.

Getting Help on Using a Function

Click a function name to open a pop-up window, and select "?" to see information on the function.

How to Create an Alias for a Query String

In graphs and charts, each variable displays the full query string in the legend. Queries are usually long and can clutter up the legend, so the alias function allows you to substitute a short name for the query string.

```
Syntax: alias(QUERY, alias)
```

Follow these steps to create an alias:

1. Create a query and select alias from the list of functions. The tag 'alias' displays in the function row and a text cursor displays inside the bracket.

2. Enter inside the brackets the alias you want to use for the variable.

To change the name again, click the text inside the brackets. Spaces are ignored but underscores can be used.
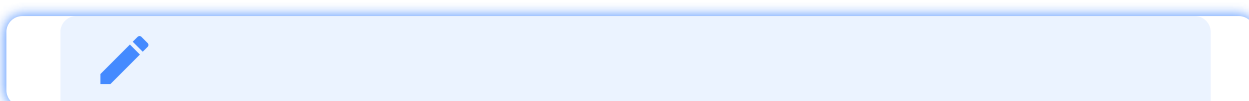
```
If you deselect the function before typing anything inside the brackets you
u will have to remove the function and add it again to rename the variabl
e.
```

Dashboard Designer Add On Functions and Syntaxes

This section describes the various functions available, when to use each function, and gives examples where helpful.

Alias

For single variable queries, renames a series to the user's input.

`Syntax: alias(QUERY, alias)`

AliasByName

For multi-variable queries, renames a series to the user's input; use it when you want to call an entire aspect. Add * as the final query in the string to return all aspect variables; in this scenario, the function will rename all variables to have the same name.

If you want to rename individual variables, select the aliasByName function from the function drop-down list. You can also stack this function with itself, using **aliasByName_stacked** which allows you to rename each variable. Here is an image that shows the 'aliasByName' function in a Query:

AliasByNamealiasByName.PNG

`Syntax: aliasByName(QUERY, name, Replacement String)`

AliasByTime

Returns grouped series under aliases based on time, and is useful for comparing data from different time periods. The input for aliasType can be weekdays, weekNumbers, months, or years. The timezoneOffset input adds a time offset from UTC in hours and can use positive or negative numbers.

`Syntax: aliasByTime(QUERY, aliasType, timezoneOffset)`

The first image below shows data returned with no AliasByTime function, followed by the same data, but grouped by the weekday of the data's timestamp:
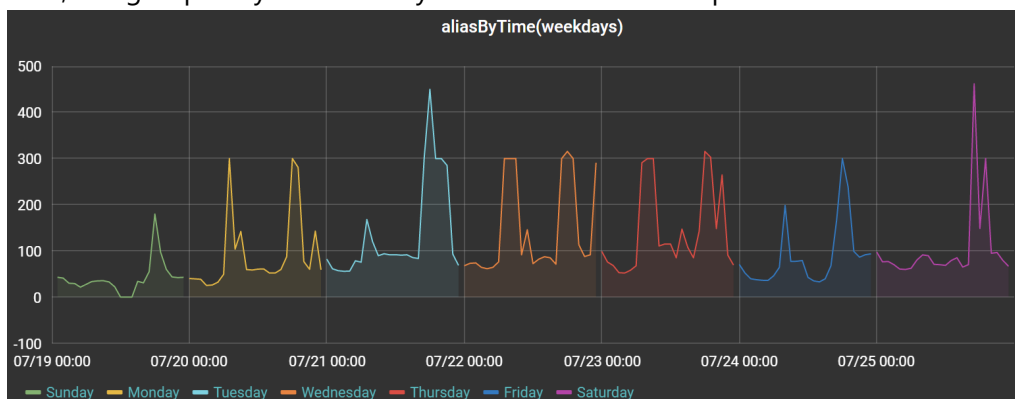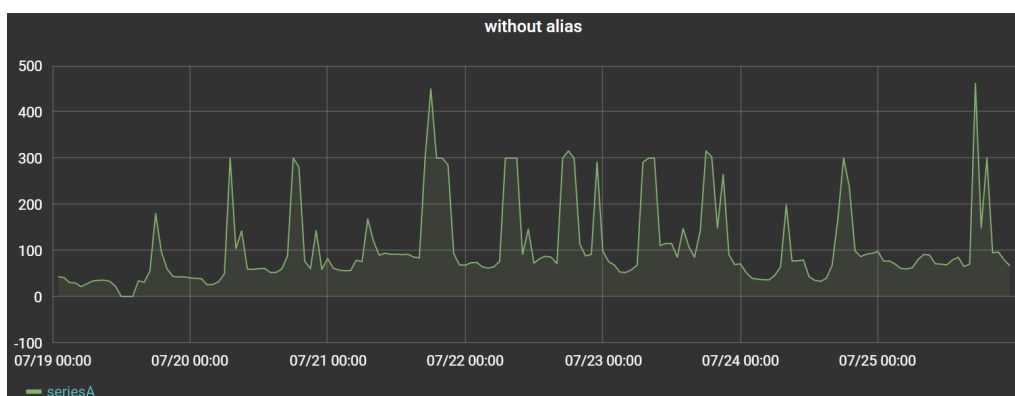
IgnoreValue

Returns queried data without the value(s) specified, and is useful for filtering out certain values from a query.

Syntax: `ignoreValue(QUERY, value)`

Value input can be:

Syntax	Meaning
`*`	wildcard - matches everything
`?`	matches any single character
`[seq]`	matches any character in seq
`[!seq]`	matches any character NOT in seq

IgnoreVariable

Returns queried data without the variable(s) specified, and is useful for filtering variables when the query has a variable wildcard e.g.Asset/Aspect/*.

Syntax: `ignoreVariable(QUERY, variable)`

Variable input can be

Syntax	Meaning
`*`	wildcard - matches everything
`?`	matches any single character
`[seq]`	matches any character in seq
`[!seq]`	matches any character NOT in seq

For example, for this list of asset names, [Motor_12, Motor_15, Motor_17, Gear, Shaft], your input would be:

```
Syntax: Motor_* -> [Motor_12, Motor_15, Motor_17]
Motor_1[27] -> [Motor_12, Motor_17]
??a* -> [Gear, Shaft]
[MG]*[!57] -> [Motor_12, Gear]
```

ℹ️

The 'ignoreVariable' function utilizes a Unix-style 'fnmatch' pattern and you can find more information here: https://docs.python.org/3/library/fnmatch.html

Math

Performs basic mathematical operations on returned variable data.

Math functions can be stacked, and they execute in the order written. Math functions apply to all data returned by the query. Here is an example:
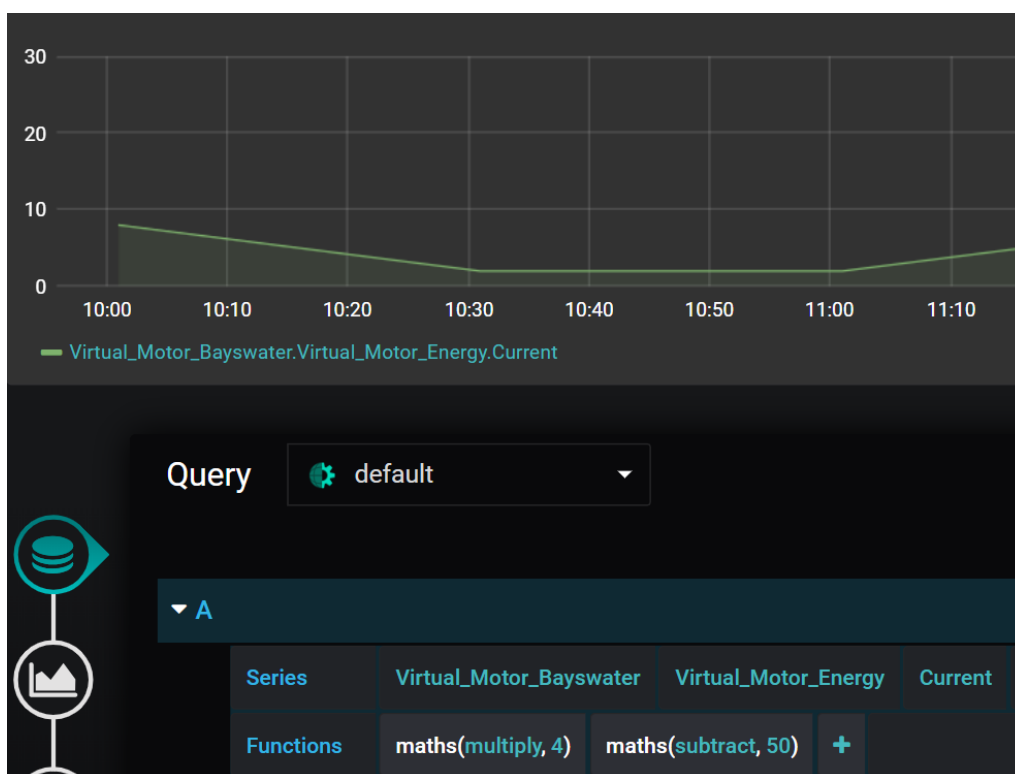
Syntax: `maths(QUERY, function, value) or math_stacked`

MSrawTimeseries

Returns raw data from a Insights Hub data source and overrides the default aggregation; instead, it returns non-aggregated or non-interpolated time series data.

When querying strings, use this function to retrieve the data, as Insights Hub's default aggregation cannot aggregate strings. This function can only retrieve up to the 2000 most-recent time series data points; a data point can contain multiple values if they share the same time index.

Syntax: `MSrawTimeseries(QUERY)`

MSassetID

Returns Insights Hub's Asset ID for the asset of the variable you select.

Syntax: `MSassetID(QUERY)`

MSmostRecentValueWithinTimeRange

Retrieves the most recent value within the queries time window using the IOT TimeSeries API. If this function does not retrieve values reliably, please use *MSaggregate(lastvalue)*.

This function retrieves the latest time index of the aspect that the variable is within. If the variable is not present in the latest time index it returns nothing. To avoid, try separating the data points you always wish to retrieve the last value of and put them in their own aspect or make sure their data is written to the latest time index.

Syntax: `MSmostRecentValueWithinTimeRange(QUERY)`

MSmostRecentValueWithinNowto90Days

Retrieves the most recent value within the last 90 days, using the IoT TimeSeries API, and regardless of the query's time window.

This function is useful for dashboards that look within a small range of time, such as a day, when you want to display the most recent value of less frequent data that does not fall within the time range. If this function does not retrieve values reliably, please use MSaggregate(lastvalue).

It retrieves the latest time index of an aspect that the variable is within. If the variable is not present in the latest time index it will return nothing. To avoid, try separating the data points. you always want to retrieve the last value of and put them in their own aspect or make sure their data is written to the latest time index.

Syntax: `MSmostRecentValueWithinNowto90Days(QUERY)`

MSboolAggregates

Returns a boolean value, according to automatically selected time ranges.

Standard Insights Hub aggregation rules cause boolean values to be misrepresented because the average aggregation rule causes the booleans to be displayed as decimals between 0 and 1.

This function aggregates the booleans and returns a boolean value by looking at the average value of booleans within the aggregated time range. If the value is below 0.5 it becomes a 0, if it is above it becomes a 1.

It allows aggregation of booleans, but still accurately represents them.

Syntax: `MSboolAggregate(QUERY)`

MSautoAggregate

Returns automatically aggregated data based on aggregation method.

Dashboard Designer Add On automatically calculates time ranges in order to automatically aggregate data using the 'average' aggregation method. This function gives power users the option to choose their own aggregation method, without having to calculate time ranges and groupings of data.

The IoT Time Series (TS) Aggregates Service creates aggregated summaries of numeric time series data and provides interfaces to read them. This allows applications to retrieve smaller datasets that cover a long time range with much better performance than processing all the raw time series data.

For example, an aspect could create new data every second, which adds up to ~2.5 million records per month. An application could use the IoT TS Aggregates Service to retrieve a summary for each day of the month, obtaining only 30 records.

Syntax: `MSautoAggregate(QUERY, method)`

MSaggregate

Returns aggregated data based on the specified intervals and method.

Dashboard Designer Add On automatically calculates time ranges to automatically aggregate data by default. This function gives users the option to fully customize how aggregates are calculated.

The IoT Time Series (TS) Aggregates Service creates aggregated summaries of numeric time series data and provides interfaces to read them. This allows applications to retrieve smaller data

sets that cover a long time range with much better performance than processing all the raw time series data.

For example, an aspect could create new data every second, which adds up to ~2.5 million records per month. An application could use the IoT TS Aggregates Service to a summary for each day of the month, obtaining only 30 records.

Syntax: `MSaggregate(QUERY, intervalUnit, intervalValue, method)`

onlyChanges

Returns only the rising and falling edges of the query.

Syntax: `onlyChanges(QUERY)`

percentageOfTimeAs

Returns the amount of time each unique value was held as a percentage of the total time range, which is useful for displaying machine states and summarizing the time elapsed in each state. This function can also stack with other functions; for example, if a variable is a string, you can stack "percentageOfTimeAs" with "MSrawTimeseries", as shown here:
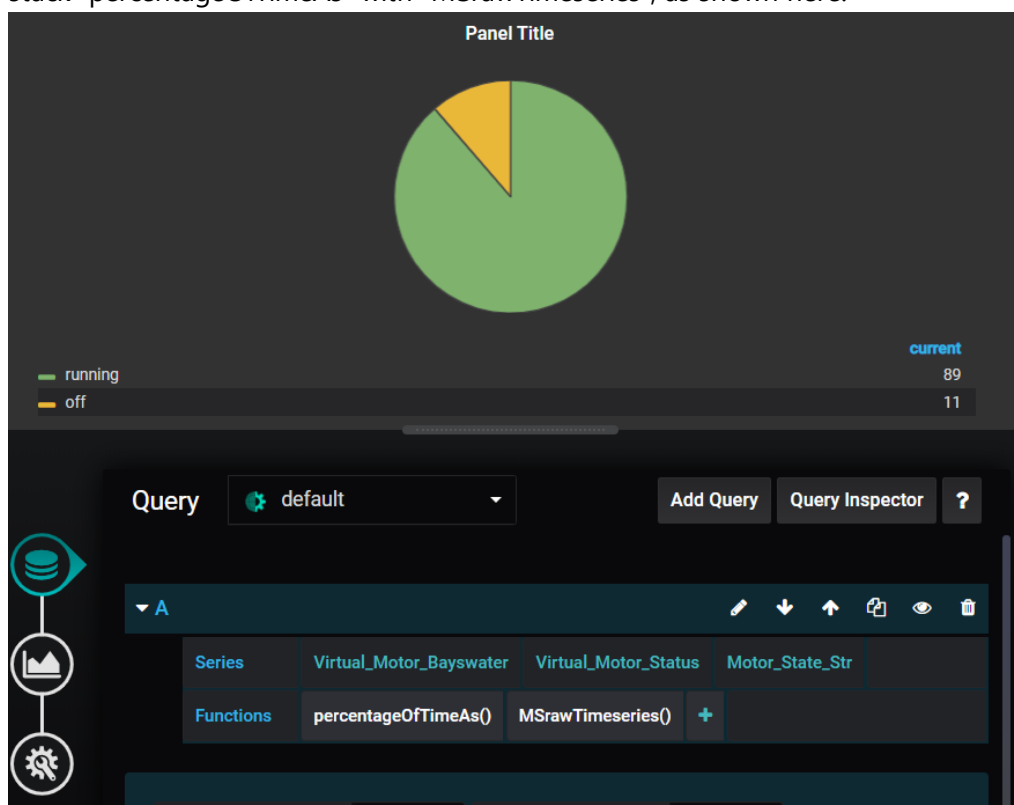
Syntax: `PercentageOfTimeAs(QUERY)`

StringToValue

Replaces strings with a number.

When displaying strings, such as in a table, you may need to change a word to a number or another descriptor, as shown here:

For example, whenever the string 'running' is found in the data it will be returned as the new designated value of '100':

Time ▾	Virtual_Motor_Bayswater.Virtual_Motor_Status.Motor_State_Str
2020-06-02 17:32:57	100.00
2020-06-02 17:29:45	off
2020-06-02 17:28:19	100.00
2020-06-02 17:26:57	100.00

Syntax: `stringToValue(QUERY, string, replacementValue)`

TimeOf

Returns the timestamp of the data points in Unix time. Use this function when you need to know the timestamp of a dataset, you can from the function.

unixTime

Making timestamps more readable in a Table panel requires that you define the datatype as 'date' in the table options. Navigate to options in the visualization tab and select 'date' from the dropdown list next to 'type'.

Time ▾	Virtual_Motor_Bayswater.Virtual_Motor_Status.Motor_State
2020-06-02 23:04:00	1.59 Tri
2020-06-02 22:08:00	1.59 Tri
2020-06-02 21:04:00	1.59 Tri
2020-06-02 17:36:00	1.59 Tri
2020-06-02 17:28:00	1.59 Tri

Because the timestamp is returned as a Unix timestamp, some additional steps may need to be carried out to make it more readable:

The results are much more readable:

Time ▾	Virtual_Motor_Bayswater.Virtual_Motor_Status.Motor_State
2020-06-02 23:04:00	2020-06-02T23:04:00+10:00
2020-06-02 22:08:00	2020-06-02T22:08:00+10:00
2020-06-02 21:04:00	2020-06-02T21:04:00+10:00
2020-06-02 17:36:00	2020-06-02T17:36:00+10:00
2020-06-02 17:28:00	2020-06-02T17:28:00+10:00

Syntax: `timeOf(QUERY)`

date_type

Displays the value in a human readable fashion.

Syntax: `date_type`

date_readable

Other panel types can also support dates (e.g. single stat). When using the timeOf function be sure to change the type or unit of the panel if it is to be displayed in a readable manner.

Syntax: `date_readable`

timeShift

Offsets a query by 86400 seconds (24 hours) by default. This is useful for overlaying data from the previous day with today's data, or overwriting the time zone of the data:
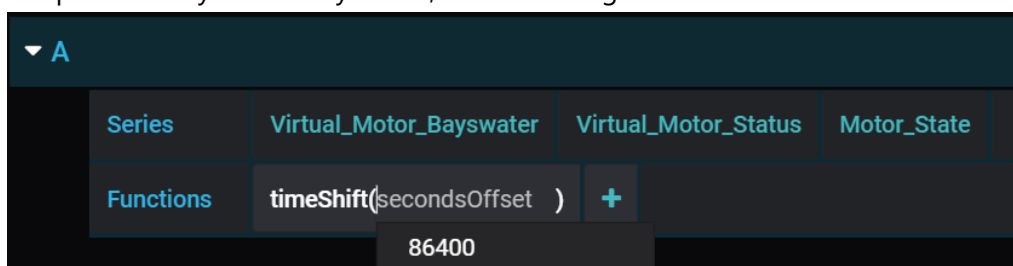
When you set up two queries for the same variable, you can overlay one with the offset and one without:
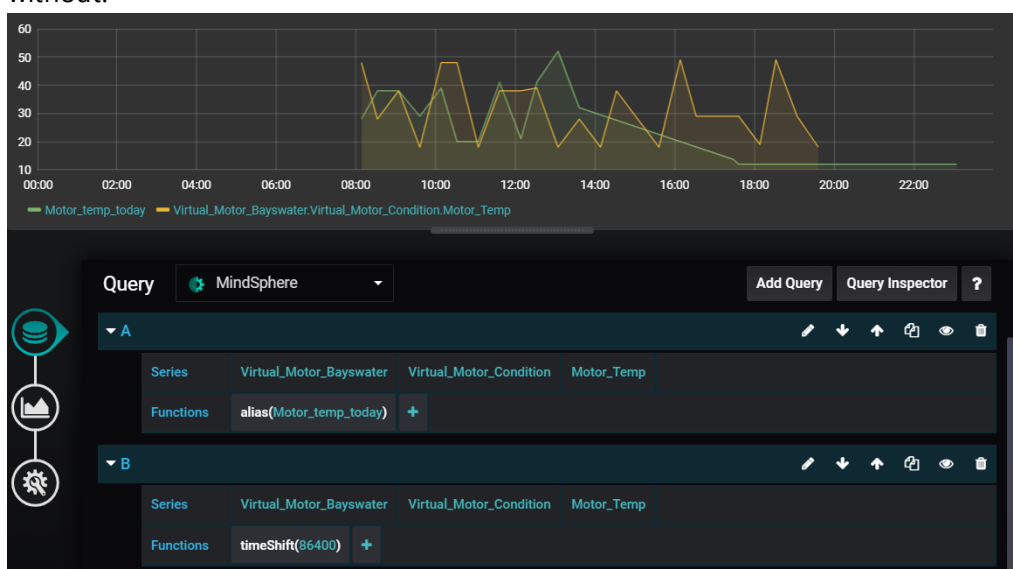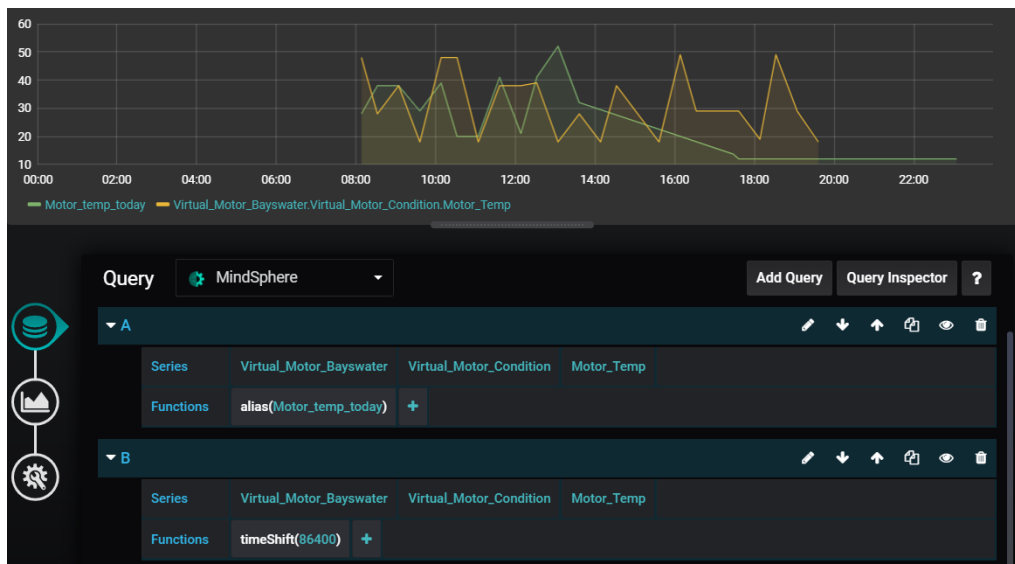
Syntax: `timeShift`

timeShift_noalias

When you notice that data displays with incorrect coloring and joining between points, this is because the graph colors and joins data based on its name. You can fix this by using the alias function to rename any variables that share the same name.

Syntax: `timeShift_noalias`

ValueToString

Replaces a value with a string. For example, in a table that displays machine state values, you can use this function to change the value to a more descriptive word:

Here is an example of a query looking for instances of "0" in the data and returning a user-selected string, "Machine_off":

Syntax: `valueToString(QUERY, value, replacementString)`

valueToString_filled

When the value '0' is found in data, it returns the new designated string, for example, 'Machine_off'. Replacement strings do not support spaces.

Syntax: `valueToString_filled(QUERY, value, replacementString)`

VFCrequest

Creates a request to the http in blocks in Visual Flow Creator (VFC).

The method parameter in the VFCrequest function defines the method of the call. The format of the endpoint parameter requires an endpoint in this format:

/public/<tenant>/<route>?key=<secret>

This function triggers a flow in VFC, and data can be returned from VFC to Dashboard Designer Add On for visualization, which requires a subscription and access to VFC.

There are also limitations with this function, including:

- Only the 'Get' method is supported and Dashboard Designer Add On selects it by default.

- Panels that use this function cannot have other queries added to the panel.

queryParams

Additional custom parameters sent to VFC as a part of the request. You can create a compatible flow in VFC by setting up the http in node as 'httpin_node'. Double click to open settings and enter the endpoint name to use.

```
Syntax: VFCrequest(method, endpoint, queryParams) httpin_endpoint
```

httpin_access

Changes the access method to public access using keys: httpin_generate, saves the flow, and copies the link address of the http in node.

httpin_key

As the endpoint parameter, this prepares the URL for use by the VFCrequest function in Dashboard Designer Add On.

When copying the URL, please remove the host URL preceding "/public".

The flow must have an "http out" at the end to cause VFC to respond, as shown in the example flows below.

Once the endpoint is created, the flow will run every time the panel refreshes.

To trigger a VFC flow that displays data in Dashboard Designer Add On, use the following JSON structure:

```
[
    {
        "datapoints": [
            [
                value,
                unix timestamp
            ],
            [
                value,
                unix timestamp
            ],
            [
                value,
                unix timestamp
            ]
        ],
        "tag": {
            "name": "variablename1"
        },
        "target": "variablename1"
    },
    {
        "datapoints": [
            [
                value,
                unix timestamp
            ],
            [
                value,
                unix timestamp
            ],
            [
                value,
                unix timestamp
            ]
        ],
        "tag": {
            "name": "variablename2"
        },
        "target": "variablename2"
    }
]
```

Here is an example that uses real data:

```
[
  {
      "datapoints": [
          [
              2384.045,
              1598382900
          ],
          [
              2387.24,
              1598383200
          ],
          [
              2414.01,
              1598383500
          ]
      ],
      "tags": {
          "name": "POWER_GENERATION"
      },
      "target": "POWER_GENERATION"
  }
]
```

Security information

Creating an http in node with a public access key means that the particular flow triggered by the http in node can potentially be accessed externally. Thus, it is important to protect your data and prevent unauthorized access. Please handle the access keys with care.

Creating an Access Control Node

Create an "access control" node after the http in node so it's possible to determine that the flow is being requested by Dashboard Designer Add On.
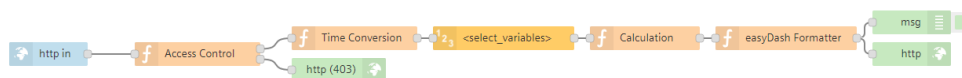
This 'access control' node, shown above, is a function node with this code:
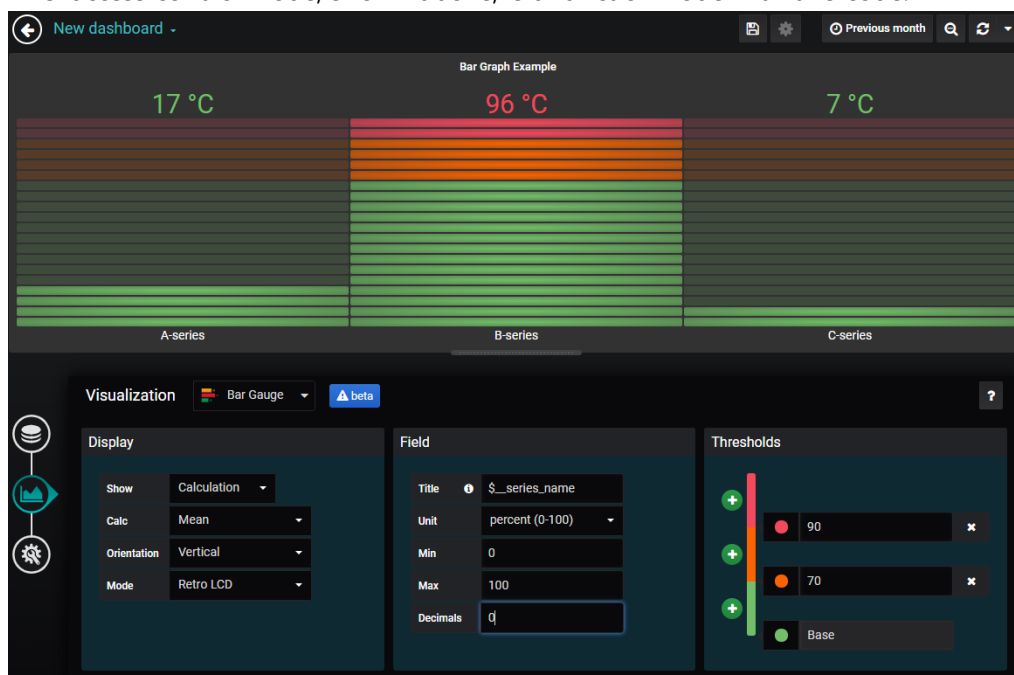
Example Flows

Follow these steps to import the example flow code:

1. Copy the code displayed below to your clipboard.

2. Open Visual Flow Creator.

3. Select "Import" and "New flow" from the app menu and paste the code.

!! note Where **https://-dashboarddesigner-datasource** appears in the flow, please substitute your tenant name; e.g., for the demo tenant, the name is https://demo-dashboarddesigner-datasource.

```
Double click the ReadMe nodes for more information.
```

Basic Flow Example

```
[
{
    "id": "6dc5e5d4.be494c",
    "type": "comment",
    "z": "392f23c2.fd446c",
    "name": "Read me",
```

```
    "info": "Access Control Node:\n    The access control node is to reduc
e the risks of using a public key.\n    This node checks the referer of th
e incoming request - if the referer does \n    not match the expected orig
in (the  Dashboard Designer Add On data source) \n    then an error 403
\"forbidden\" will be returned to the requester.\n    \n    \n\nTime Conve
rsion Node:\n    Converts Dashboard Designer Add On timestamps to the form
at that VFC uses \n    \nRead Timeseries Node:\n    Select the variable(s)
you wish to manipulate and display. \n    Make sure that the \"Mode\" is s
et to \"Interval\" and that the interval fields \n    are left blank if yo
u want the time range to match that of the dashboard \n    making the requ
est.\n    \n\"Calculation\" Function Node:\n    The Calculation node in th
is example flow is a Function node and has a very\n    simple calculation
- it divides all the values by 2.\n    \n\"Dashboard Designer Add On Forma
tter\" Function Node:\n    This is an example node which provides a method
of converting a typical\n    MindSphere response and a typical analytics n
ode response. \n    If custom analytics functions are being implemented th
en it is likely that\n    this node will need adjustment.",
    "sticky": 0,
    "x": 340,
    "y": 1200,
    "wires": [],
    "_type": "node"
},
{
    "id": "11c225e8.96268a",
    "type": "debug",
    "z": "392f23c2.fd446c",
    "name": "",
    "active": true,
    "console": "false",
    "xaxis": "_time",
    "complete": "true",
    "x": 1250,
    "y": 1220,
    "wires": []
},
{
    "id": "5bc50450.2a2fec",
    "type": "function",
    "z": "392f23c2.fd446c",
    "name": "Access Control",
    "func": "let refererURL = \"https://<tenantname>-dashboarddesigner-dat
asource\";\nif(msg.req.headers.referer == refererURL){\nreturn [msg,null];
```

```
\n}else{\n    console.log(\"Forbidden\");\n    msg = {};\n    msg.statusCo
de = 403;\n    \n    return [null,msg];\n}\n",
    "outputs": "2",
    "noerr": 0,
    "x": 340,
    "y": 1260,
    "wires": [
        [
            "e8fdc546.0c8628"
        ],
        [
            "37466b30.e8dfa4"
        ]
    ]
},
{
    "id": "e8fdc546.0c8628",
    "type": "function",
    "z": "392f23c2.fd446c",
    "name": "Time Conversion",
    "func": "msg.from = new Date(parseInt(msg.req.query.from));\nmsg.to =
new Date(parseInt(msg.req.query.to));\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "x": 540,
    "y": 1240,
    "wires": [
        [
            "285dd047.2206f"
        ]
    ]
},
{
    "id": "f09ef087.5c316",
    "type": "function",
    "z": "392f23c2.fd446c",
    "name": "Calculation",
    "func": "for (let i in msg.payload){ \n    for (let key in msg.payload
[i]) {\n        if (key != \"_time\") {\n            msg.payload[i][key] =
(msg.payload[i][key]/2);\n        }\n    }\n}\n\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "x": 910,
```

```
            "y": 1240,
            "wires": [
                [
                    "fe227dcc.84658"
                ]
            ]
        },
        {
            "id": "fe227dcc.84658",
            "type": "function",
            "z": "392f23c2.fd446c",
            "name": "Dashboard_Designer Formatter",
            "func": "let msVariables = {};\n\n// this function makes a number of a
ssumptions about the data coming in and was written for data outputs of th
e VFC analytics nodes\nif(!Array.isArray(msg.payload)){ // this function c
heck to see if the payload is an array, which is required for the Dashboar
d_Designer formatting\n    console.log(\"payload is not the correct forma
t, attempting to readjust\"); \n    if(typeof(msg.payload)!=='object' && m
sg.parameter){\n        msg.payload = { [msg.parameter]: msg.payload };\n
    }\n    if(!msg.payload._time){ // assuming there is only one data point in
the payload that triggers this function - typical outputs of analytics nod
es will have one value and no timestamp - this ensures the data will popul
ate the panel in Dashboard_Designer\n        msg.payload._time = msg.to;\n
    }\n    msg.payload = [msg.payload]; // once the above checks have been mad
e we must make an array for the next function to execute properly.\n}\nif
(Array.isArray(msg.payload)){\n    for (let i in msg.payload){ \n         f
or (let key in msg.payload[i]) {\n            if (key != \"_time\") {\n
if (!msVariables[key]) {\n                 msVariables[key] = { \"datap
oints\": [], \"tag\": {\"name\": key}, \"target\": key }; //our Dashboard_
Designer object\n                 }\n                 msVariables[key].datap
oints.push([msg.payload[i][key],new Date(msg.payload[i]._time).getTime()/1
000]);\n            }\n        }\n    }\n    msg.payload = [];\n    for (l
et i in msVariables) {\n        msg.payload.push(msVariables[i]);\n    }\n
return msg;\n}else{\n    console.log(\"Data could not be formatted correct
ly\");\n}\n",
            "outputs": 1,
            "noerr": 0,
            "x": 1090,
            "y": 1240,
            "wires": [
                [
                    "11c225e8.96268a",
                    "3c3e8ca3.ac8b14"
```

```
        ]
    ]
},
{
    "id": "9368a33c.14e98",
    "type": "http in",
    "z": "392f23c2.fd446c",
    "name": "http in",
    "endpoint": "",
    "method": "get",
    "upload": false,
    "access": "private",
    "key": "",
    "users": "",
    "x": 170,
    "y": 1260,
    "wires": [
        [
            "5bc50450.2a2fec"
        ]
    ]
},
{
    "id": "3c3e8ca3.ac8b14",
    "type": "http response",
    "z": "392f23c2.fd446c",
    "name": "",
    "statusCode": "",
    "headers": {},
    "x": 1250,
    "y": 1260,
    "wires": []
},
{
    "id": "37466b30.e8dfa4",
    "type": "http response",
    "z": "392f23c2.fd446c",
    "name": "",
    "statusCode": "403",
    "headers": {},
    "x": 520,
    "y": 1280,
    "wires": []
```

```
    },
    {
        "id": "285dd047.2206f",
        "type": "read timeseries",
        "z": "392f23c2.fd446c",
        "name": "",
        "topic": "",
        "topicLabel": "",
        "assetName": "",
        "period": "60",
        "offset": "0",
        "mode": "interval",
        "from": "",
        "datetimepickerFrom": "",
        "to": "",
        "datetimepickerTo": "",
        "timezoneoffset": 0,
        "x": 720,
        "y": 1240,
        "wires": [
            [
                "f09ef087.5c316"
            ]
        ]
    }
    ]
```

Analytics Example: Moving Average

```
 [
{
    "id": "693080.833c5f8",
    "type": "comment",
    "z": "392f23c2.fd446c",
    "name": "Read me",
    "info": "Access Control Node:\n    The access control node is to reduc
e the risks of using a public key.\n    This node checks the referer of th
e incoming request - if the referer does \n    not match the expected orig
in (the Dashboard_Designer data source) \n    then an error 403 \"forbidde
n\" will be returned to the requester.\n\n \n\nTime Conversion Node:\n
Converts Dashboard_Designer timestamps to the format that VFC uses \n
\nRead Timeseries Node:\n    Select the variable(s) you wish to manipulate
and display. \n    Make sure that the \"Mode\" is set to \"Interval\" and
that the interval fields \n    are left blank if you want the time range t
```

o match that of the dashboard \n   making the request.\n   \n\"Dashboard
_Designer Formatter\" Function Node:\n   This is an example node which pr
ovides a method of converting a typical\n   MindSphere response and a typ
ical analytics node response. \n   If custom analytics functions are bein
g implemented then it is likely that\n   this node will need adjustmen
t.",
    "sticky": 0,
    "x": 320,
    "y": 1440,
    "wires": [],
    "_type": "node"
},
{
    "id": "4dea3605.6899e8",
    "type": "debug",
    "z": "392f23c2.fd446c",
    "name": "",
    "active": true,
    "console": "false",
    "xaxis": "_time",
    "complete": "true",
    "x": 1230,
    "y": 1460,
    "wires": []
},
{
    "id": "2ec0625a.b5ad7e",
    "type": "function",
    "z": "392f23c2.fd446c",
    "name": "Access Control",
    "func": "let refererURL = \"https:// Dashboard_Designer-datasource-sil
opsms.apps.eu1.mindsphere.io\";\nif(msg.req.headers.referer == refererURL)
{\nreturn [msg,null];   \n}else{\n   console.log(\"Forbidden\");\n   ms
g = {};\n   msg.statusCode = 403;\n   \n   return [null,msg];\n}\n",
    "outputs": "2",
    "noerr": 0,
    "x": 320,
    "y": 1500,
    "wires": [
        [
            "aee3ad9b.e4ad"
        ],
        [
```

```
                "62e6b5bb.c58bcc"
            ]
        ]
    },
    {
        "id": "aee3ad9b.e4ad",
        "type": "function",
        "z": "392f23c2.fd446c",
        "name": "Time Conversion",
        "func": "msg.from = new Date(parseInt(msg.req.query.from));\nmsg.to =
new Date(parseInt(msg.req.query.to));\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "x": 520,
        "y": 1480,
        "wires": [
            [
                "8ab9a317.bcd2a"
            ]
        ]
    },
    {
        "id": "f0f3fae1.aff638",
        "type": "function",
        "z": "392f23c2.fd446c",
        "name": "Dashboard_Designer Formatter",
        "func": "let msVariables = {};\n\n// this function makes a number of a
ssumptions about the data coming in and was written for data outputs of th
e VFC analytics nodes\nif(!Array.isArray(msg.payload)){ // this function c
heck to see if the payload is an array, which is required for the Dashboar
d_Designer formatting\n    console.log(\"payload is not the correct forma
t, attempting to readjust\"); \n    if(typeof(msg.payload)!=='object' && m
sg.parameter){\n        msg.payload = { [msg.parameter]: msg.payload };\n
}\n    if(!msg.payload._time){ // assuming there is only one data point in
the payload that triggers this function - typical outputs of analytics nod
es will have one value and no timestamp - this ensures the data will popul
ate the panel in Dashboard_Designer\n        msg.payload._time = msg.to;\n
}\n    msg.payload = [msg.payload]; // once the above checks have been mad
e we must make an array for the next function to execute properly.\n}\nif
(Array.isArray(msg.payload)){\n    for (let i in msg.payload){ \n        f
or (let key in msg.payload[i]) {\n            if (key != \"_time\") {\n
if (!msVariables[key]) {\n                    msVariables[key] = { \"datap
oints\": [], \"tag\": {\"name\": key}, \"target\": key }; //our Dashboard_
```

```
Designer object\n                    }\n                    msVariables[key].datap
oints.push([msg.payload[i][key],new Date(msg.payload[i]._time).getTime()/1
000]);\n              }\n           }\n        }\n    msg.payload = [];\n    for (l
et i in msVariables) {\n          msg.payload.push(msVariables[i]);\n      }\n
return msg;\n}else{\n    console.log(\"Data could not be formatted correct
ly\");\n}\n",
    "outputs": 1,
    "noerr": 0,
    "x": 1070,
    "y": 1480,
    "wires": [
        [
            "4dea3605.6899e8",
            "ee0fa9f.f331758"
        ]
    ]
},
{
    "id": "b19229df.5310f8",
    "type": "http in",
    "z": "392f23c2.fd446c",
    "name": "",
    "endpoint": "",
    "method": "get",
    "upload": false,
    "access": "private",
    "key": "",
    "users": "",
    "x": 170,
    "y": 1500,
    "wires": [
        [
            "2ec0625a.b5ad7e"
        ]
    ]
},
{
    "id": "ee0fa9f.f331758",
    "type": "http response",
    "z": "392f23c2.fd446c",
    "name": "",
    "statusCode": "",
    "headers": {},
```

```
        "x": 1230,
        "y": 1500,
        "wires": []
    },
    {
        "id": "62e6b5bb.c58bcc",
        "type": "http response",
        "z": "392f23c2.fd446c",
        "name": "",
        "statusCode": "403",
        "headers": {},
        "x": 500,
        "y": 1520,
        "wires": []
    },
    {
        "id": "63fcd020.91557",
        "type": "moving average",
        "z": "392f23c2.fd446c",
        "name": "",
        "parameter": "",
        "parameterout": "",
        "windowSize": 3,
        "algorithm": "simple",
        "alpha": 0.5,
        "x": 880,
        "y": 1480,
        "wires": [
            [
                "f0f3fae1.aff638"
            ]
        ]
    },
    {
        "id": "8ab9a317.bcd2a",
        "type": "read timeseries",
        "z": "392f23c2.fd446c",
        "name": "",
        "topic": "",
        "topicLabel": "",
        "assetName": "",
        "period": "60",
        "offset": "0",
```

```json
            "mode": "interval",
            "from": "",
            "datetimepickerFrom": "",
            "to": "",
            "datetimepickerTo": "",
            "timezoneoffset": 0,
            "x": 700,
            "y": 1480,
            "wires": [
                [
                    "63fcd020.91557"
                ]
            ]
        }
    ]
```

Access Query Parameters Sent to VFC

```json
 [
{
        "id": "3703d4c5.4b51ac",
        "type": "comment",
        "z": "392f23c2.fd446c",
        "name": "Read me",
        "info": "This flow provides an example of how the query parameters can
be accessed from within a Visual Flow Creator flow.",
        "sticky": 0,
        "x": 540,
        "y": 1600,
        "wires": [],
        "_type": "node"
},
{
        "id": "1edecaa1.50b9b5",
        "type": "function",
        "z": "392f23c2.fd446c",
        "name": "Access Control",
        "func": "let refererURL = \"https:// Dashboard_Designer-datasource-sil
opsms.apps.eu1.mindsphere.io\";\nif(msg.req.headers.referer == refererURL)
{\nreturn [msg,null];    \n}else{\n    console.log(\"Forbidden\");\n    ms
g = {};\n    msg.statusCode = 403;\n    \n    return [null,msg];\n}\n",
        "outputs": "2",
        "noerr": 0,
        "x": 551.5,
```

```
            "y": 1661,
            "wires": [
                [
                    "3f44162d.f2f52a"
                ],
                [
                    "fcfa3a54.111b18"
                ]
            ]
        },
        {
            "id": "3f44162d.f2f52a",
            "type": "function",
            "z": "392f23c2.fd446c",
            "name": "debug query param",
            "func": "console.log(msg.req.query.exampleQueryParam);\n\nreturn ms
g;",
            "outputs": 1,
            "noerr": 0,
            "x": 761.5,
            "y": 1641,
            "wires": [
                []
            ]
        },
        {
            "id": "e9b94d90.5d4a6",
            "type": "http in",
            "z": "392f23c2.fd446c",
            "name": "",
            "endpoint": "",
            "method": "get",
            "upload": false,
            "access": "private",
            "key": "",
            "users": "",
            "x": 381.5,
            "y": 1661,
            "wires": [
                [
                    "1edecaa1.50b9b5"
                ]
            ]
```

```
},
{
    "id": "fcfa3a54.111b18",
    "type": "http response",
    "z": "392f23c2.fd446c",
    "name": "",
    "statusCode": "403",
    "headers": {},
    "x": 731.5,
    "y": 1681,
    "wires": []
}
 ]
```

string_table

Note: Replacement strings do not support spaces.

VFCrequest

Creates a request to the http in blocks in Visual Flow Creator (VFC). Data can be returned from VFC to Dashboard Designer for visualization. This requires a subscription and access to Visual Flow Creator.

Important Points About VFC Requests:

- Only the GET method is currently supported and is selected by default.

- A panel that uses VFCrequest cannot have additional queries in the same panel.

- No asset query is required for this function, as the data is retrieved via Visual Flow Creator.

- The method parameter defines the method of the call.

The format of the endpoint parameter requires an endpoint which follows this format: `/public/<tenant>/<route>?key=<secret>`.

Syntax: `VFCrequest(method, endpoint, queryParams)`

queryParams

Additional custom parameters that can be sent to VFC as a part of the request; this creates a compatible flow in Visual Flow Creator when you set up the http in node in Visual Flow Creator.

httpin_node

Double click to open the settings and enter the endpoint name that you wish to use.

8.1 Dashboard Designer Add On Functions

httpin_endpoint

Changes the Access method to Public access using keys.

httpin_generate

Save this flow and copy the link address of the http in node.

httpin_key

The URL used by the VFCrequest function as the endpoint parameter. If you copied the URL
manually, be sure to remove the host URL preceding /public. For Visual Flow Creator to respond,
the flow should have an http out at the end as seen in the example flows below.
Once the endpoint is created, the flow will run every time the panel refreshes, as long as it is
formatted correctly. Dashboard Designer Add On can only display the data configured with the
correct JSON structure, as shown in the example below.

Security Information

Creating an http in node with a public access key means that the particular flow triggered by the
http in node can potentially be accessed externally. Thus, it is important to protect your data and
prevent unauthorized access. Please handle the access keys with care.
It is recommended to create an "access control" node after the http in node so that the flow can
determine if the flow is being requested by Dashboard Designer Add On.
The "access control" node shown above is a function node with the code:
Importing an Example Flow
To import an example flow from below, copy the code and select Import > new flow from the
app menu in Visual Flow Creator. Double-click the Read me nodes for more information.

Example: Access Query Parameters Sent to VFC

```
[
{
    "id": "3703d4c5.4b51ac",
    "type": "comment",
    "z": "392f23c2.fd446c",
    "name": "Read me",
    "info": "This flow provides an example of how the query parameters can
be accessed from within a Visual Flow Creator flow.",
    "sticky": 0,
    "x": 540,
    "y": 1600,
    "wires": [],
    "_type": "node"
```

```
    },
    {
        "id": "1edecaa1.50b9b5",
        "type": "function",
        "z": "392f23c2.fd446c",
        "name": "Access Control",
        "func": "let refererURL = \"https://Dashboard_Designer-datasource-silo
psms.apps.eu1.mindsphere.io\";\nif(msg.req.headers.referer == refererURL)
{\nreturn [msg,null];     \n}else{\n    console.log(\"Forbidden\");\n    ms
g = {};\n    msg.statusCode = 403;\n    \n    return [null,msg];\n}\n",
        "outputs": "2",
        "noerr": 0,
        "x": 551.5,
        "y": 1661,
        "wires": [
            [
                "3f44162d.f2f52a"
            ],
            [
                "fcfa3a54.111b18"
            ]
        ]
    },
    {
        "id": "3f44162d.f2f52a",
        "type": "function",
        "z": "392f23c2.fd446c",
        "name": "debug query param",
        "func": "console.log(msg.req.query.exampleQueryParam);\n\nreturn ms
g;",
        "outputs": 1,
        "noerr": 0,
        "x": 761.5,
        "y": 1641,
        "wires": [
            []
        ]
    },
    {
        "id": "e9b94d90.5d4a6",
        "type": "http in",
        "z": "392f23c2.fd446c",
        "name": "",
```

```
        "endpoint": "",
        "method": "get",
        "upload": false,
        "access": "private",
        "key": "",
        "users": "",
        "x": 381.5,
        "y": 1661,
        "wires": [
            [
                "1edecaa1.50b9b5"
            ]
        ]
    },
    {
        "id": "fcfa3a54.111b18",
        "type": "http response",
        "z": "392f23c2.fd446c",
        "name": "",
        "statusCode": "403",
        "headers": {},
        "x": 731.5,
        "y": 1681,
        "wires": []
    }
        ]
```

Visualizations and Plugins 9

9.1 Visualizations in Dashboard Designer Add On

Dashboard Designer Add On's visualization options allow you to get the most out of the data you want to visually analyze. This section includes overviews and details on visualization options and features.

> ✏️
>
> While this topic covers all Dashboard Designer Add-On visualization panels, please note that not all visualization panels are supported in LPC subscriptions.

About Functions

Functions associate or transform raw data into useful outputs that help you create meaningful dashboards. When queries request data, the data is automatically aggregated in a format similar to other Insights Hub Monitor apps.

Default Query Settings Overrides

When you want to query data in a way that differs from the default query settings, you can use functions to override the default query settings. See the Functions topic for more information.

Graph

Allows for the representation of multiple values over a period of time or as a histogram. This graph also displays data point details when you hover over a specific point:
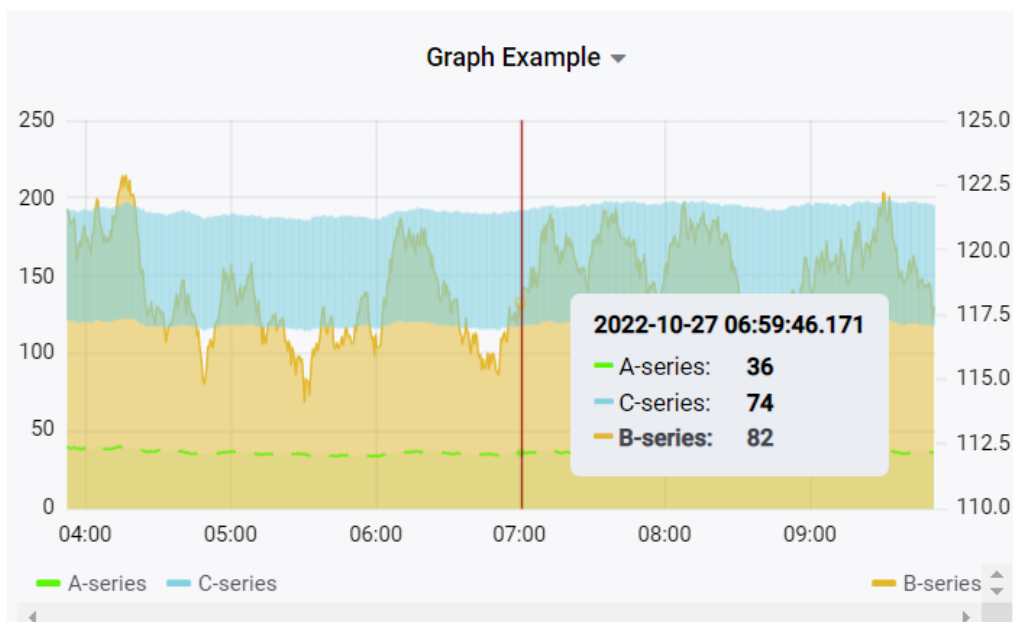
Draw Modes

Bars displays the data as a bar chart.

Lines displays the data as a line graph.

Points displays the data as points on a graph.

Mode Options

Fill fills the area between the graph and the X- axis. (0 = no fill, 10 = full fill).

Line Width allows you to specify the width of the lines that display in a graph.

Staircase displays the data as a stepped line graph.

Hover Tooltip Modes

All Series displays graph series as a tool tip when you hover over graph data.

Single displays a single series as a tool tip when you hover over graph data.

Sort Order requires you to select "All Series" first, then you can change the tool tip order to 'none' which, in turn, allows the query's setting to prevail and the order, ascending or descending, is determined by what you select.

Stacking & Null Values

Stack enables data queries to stack on top of one another.

Null Value allows you to choose how null values display.

Connected skips null values and connects the graph's line to the next non-null value.

Null creates a gap in the line graph.

Null as zero renders null values as zeroes.

Series Override

Allows the selected series to display in a style different from the other elements.

Axes

Show displays the chosen axis. To assign data to the right Y-axis, use the Series Override function.
Unit allows you to select a unit for the axis data.
Scale changes the scale of the axis from Linear to log (base 2), log (base 10), log (base 32) and log (base 1024).
Y-Min & Y-Max sets the minimum of maximum value for the axis.
Decimals sets the number of decimals to display for the axis data.
Label displays labels on the axes.

X-Axis Mode

Time displays the data with reference to time.
Series displays the data grouped by series, and allows changing the type of data represented by selecting from the drop-down list options, avg, min, max, total, count, and current.
Histogram groups values as ranges, and allows customizing the number of ranges (buckets) to display, and filters to leave out specified min and max values.

Legend Options

Show displays the graph legend with the series name and color.
As table displays the legend as a table.
To the right displays the legend to the right of the graph.
Width sets the minimum width for the table (in pixels).
Values displays toggled values in the legend.
Hide series hides series that return only nulls or zeros.

Threshold & Time Regions

Threshold overlays visual thresholds on the graph to indicate greater than (gt) or less than (lt) the described value.
You can customize the threshold color, fill, line and which y-axis the threshold applies to.
Time Region overlays a visual indication of a time range (different days of the week); to time series data.

> ✏️
>
> Raise or lower the threshold value by dragging the threshold value displayed on the right side of the graph.

Singlestat

Displays the value of the queried data and only displays a single query.
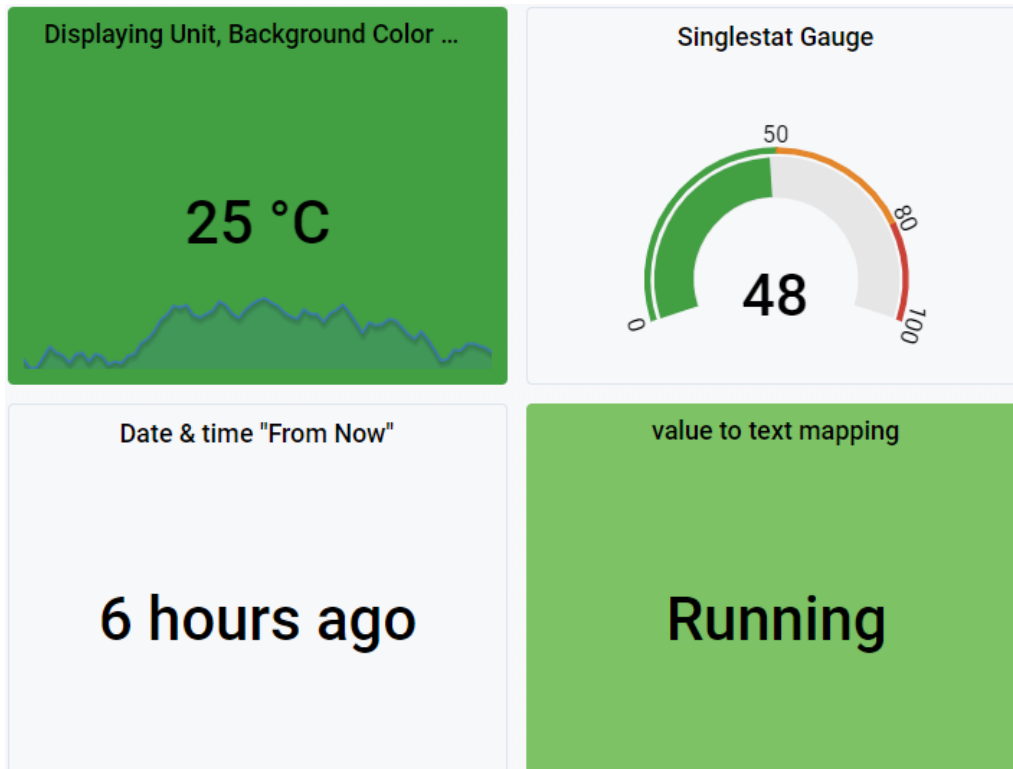
Value displays the data according to the option selected within the selected time range. Includes Min, Max, Average, Current (latest), Total, Name (of series), First, Delta (total incremental increase (of a counter) in the series), Difference, Range and Time (stamp) of last point.

Prefix adds a prefix to the value.

Postfix adds a postfix to the value. Useful for custom units.

Unit adds a unit to the value, before the postfix.

Decimals sets the number of decimals that display.

Coloring

Background adds the selected color to the background of the panel.

Value adds the selected color to the value text.

Prefix adds the selected color to the prefix text.

Postfix adds the selected color to the postfix text.

Thresholds sets the thresholds for colored visualization.

Colors sets the colors that the threshold values represent.

Spark lines adds a basic line graph to the panel with custom visualization options available.

Gauge adds a gauge visualization behind the single stat with custom visualization options available.

Value Mappings allow values, or a range of values, to display as text.

!!! note If you want the visualization of a Boolean value to render in color, set the threshold value to "0.1, 0.9", which changes the first and last color to the color you specify.

Gauge

Displays a gauge with slightly different options compared to Singlestat.

All Values displays a separate gauge for each row of data, and the "Limit" input restricts it to a maximum number of rows.

Calculation displays a calculated value within the chosen data range.

Calculations you can select include:

- Last (current value)

- Minimum

- Maximum

- Mean

- Total count

- Delta (total incremental increase (of a counter) in the series)

- Step (minimum interval between values)

- Difference

- Min (above zero)

- Change count (number of times the value changes)

- Distinct count (number of unique values)

Labels displays the gauge labels that are set in the "Thresholds" section

Markers displays the gauge color markers that are set in the "Thresholds" section)

Field Title adds a title or description beneath the value

Unit selects the unit of measure for the value

Min and Max select the minimum and maximum values for the gauge

Decimals sets the number of decimals to display

Thresholds sets the threshold value and associated color

Value Mappings allows for values (or a range of values) to display as text

Bar Gauge

Displays single data points in bars.
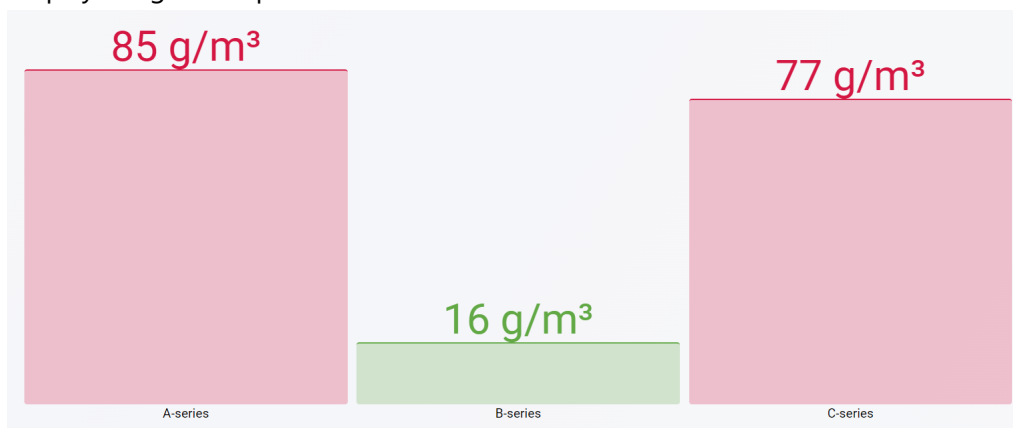
Show All Values displays a separate gauge for each row of data, the "Limit" input keeps it to a maximum number of rows

Calculation displays a calculated value within the chosen data range. Calculations you can select include:

- Last (current value)

- First

- Minimum

- Maximum

- Mean

- Total count

- Delta (total incremental increase (of a counter) in the series

- Step (minimum interval between values)

- Difference

- Min (above zero)

- Change count (number of times the value changes)

- Distinct count (number of unique values)

Bar Gauge Display

Orientation sets the bars to display horizontally or vertically

Mode selects the way to visually represent the bar gauge: Basic, Retro LED, or Gradient.

Fields

Title ads a title and description above the bar gauge

Unit select the unit for the value

Min and Max select the minimum and maximum values for the gauge

Decimals sets the number of decimals to display

Threshold sets the threshold value and the associated color

Value Mappings allows for values (or a range of values) to display as text

Breadcrumb

Breadcrumb is a panel plugin that tracks dashboards you visit during a session and displays them as a breadcrumb.

Dashboards are listed once even if you visit them more, and the names are hyperlinked for easy returns. Navigating back to a dashboard resets any navigation movements made after the dashboard you return to.

!!! note Breadcrumbs only track dashboards that include a Breadcrumb panel.

Is Root

Is Root sets the current dashboard as the root and clears breadcrumbs in the root dashboard.

Is root dashboard clears the breadcrumb in the root dashboard.

Hide text in root dashboard displays nothing if the breadcrumb has only one item.

Limit the amount of breadcrumb items to limits the number of breadcrumb items and removes the oldest item when the number of items is surpassed.

Limit sets a limit on breadcrumb items. When the item limit is reached, the oldest item drops from the breadcrumb when a new one is added.

Table

Displays data in a table format.

Table Data

Table Transform sets the type of transformation(s) required:

Time Series to rows automatically sets columns as Time, Series Name, and Value.

Time Series to Column automatically sets columns as Time and Value.

Time Series Aggregations automatically sets columns as Series Name and Average, although you can substitute Min, Max, Avg, Total, Current (last value), or Count for Average.

JSON Data displays raw value data and timestamp data in JSON format.

Paging offers options for large amounts of data, like limiting the number of rows per page, adding page scrolling, and changing font size.

Column Styles

All column styles are editable. Use the:

Options identify the specific column to change by using the "Apply to columns named" input. You can change the column title (header) now, and, if you want, render the value as a link to click through.

Type depending on the column content data type, i.e., number, string, date, hidden, you can customize thresholds, apply colors to values, cells, and rows, and add value mapping, which transforms values or value ranges to text.

Text

Text is a simple panel that displays text that you can add using MarkDown or HTML formats.

Heatmap
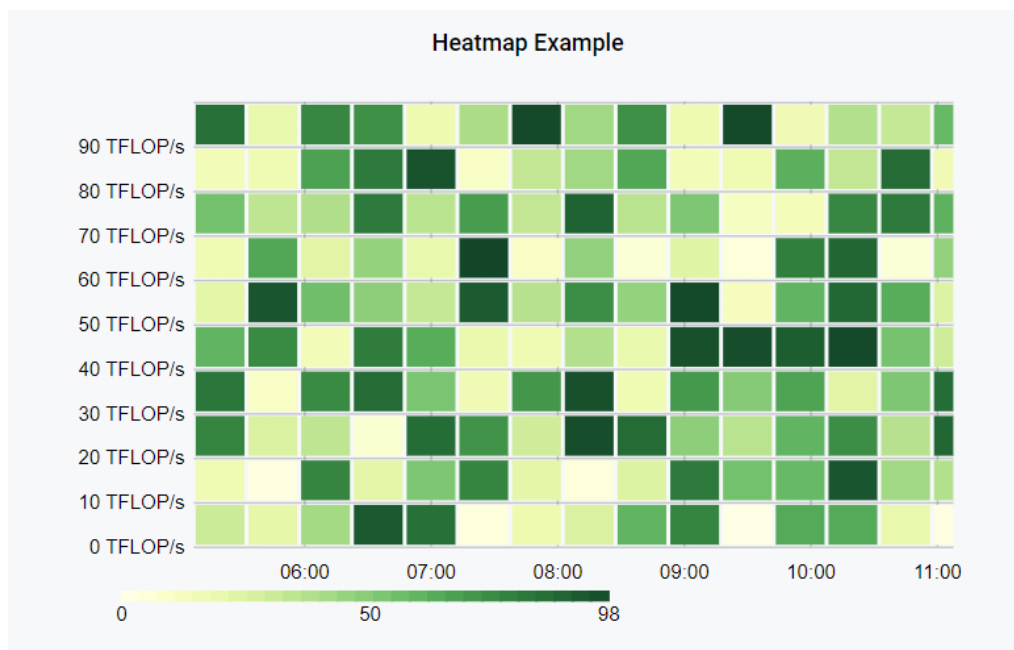
Y-Axis

Unit: sets the unit for axis values.

Scale changes the axis scale from Linear to log (base 2), log (base 10), log (base 32) and log (base 1024).

Y-min & Y-max sets the minimum of maximum values for the axis

Decimals: sets the number of decimal placess to display for axis values.

Buckets

You can change both the number and size of data buckets according to visualization requirements for both the Y- and X-axes.

Data Format

You can change between time series and time series buckets.

Display

Numerous visualizations are available, for example, color mode (spectrum or opacity), color scale, show/hide legend, bucket visualizations, bucket spacing and shape, and tool tip display.

Dashboard List

Dashboard List Options

Starred displays favorite dashboards

Recently Viewed displays recently viewed dashboards

Search allows users to search dashboards from a panel

Show Headings displays dashboard headings

Max Items displays the input amount of dashboards

Search when "search" is enabled, dashboard display can be limited to set queries, folders or tags

Clock
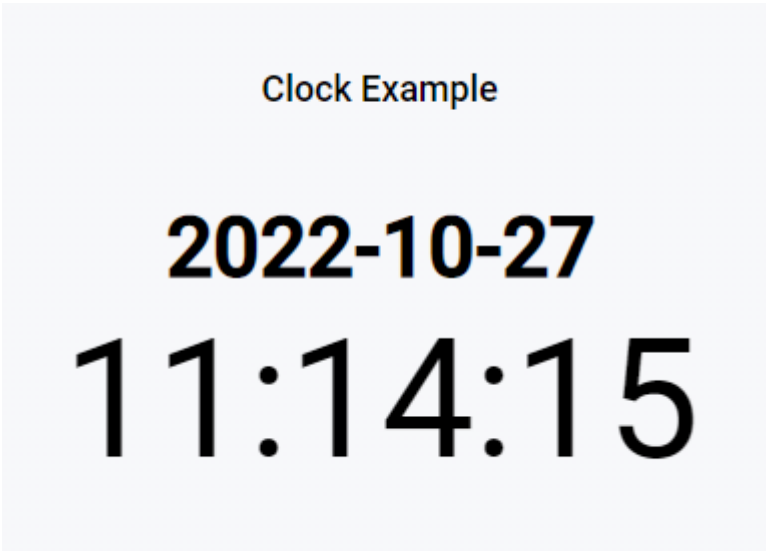
Clock Options

Clock Mode sets the clock to show time or a countdown.

Background color sets the background color of the clock panel.

Countdown sets an action to trigger when the 'countdown' completes, like generating a text message.

Time format when you select "time", additional options appear, including changing between 12 and 24 hours and changing the font size and weight.

Time Zone in both countdown and time clock modes, you can change the time zone display. Your selections here override settings specified for the dashboard. To change your dashboard's time zone, navigate to the dashboard and click the configure icon (upper right) and select the time zone (default, local or UTC in the General settings.

Refresh interval enables the Sync option so the clock synchronizes each time the page refreshes.Integrated Data Lake (IDL) widgets can refresh as quickly as 15 seconds; however, a refresh rate below 15 seconds for IDL widgets could result in canceled queries.

Pie Chart

Displays bucket data in a pie or donut chart.
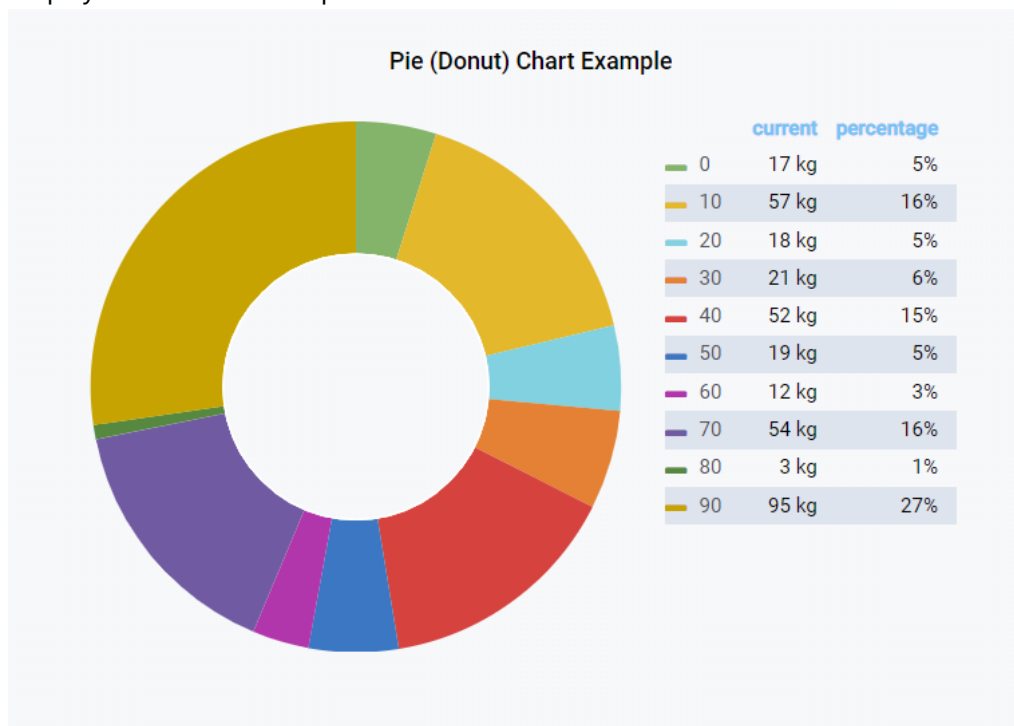
Unit specifies a unit for pie chart values.

Value sets the value or calculation to apply to data from current (last value), minimum, maximum, average and total.

Divider width sets the width in pixels between pie chart segments.

Pie Chart Legend

Show Legend dipslays the legend on the panel.

Position sets the position of the legend relative to the pie chart.

Legend Breakpoint/ Width sets the relative size of the legend relative to its position on the panel.

Legend Values displays the values of the Pie Chart.

Values header overrides the name of the value column.

Values decimals sets the number of decimals to display.

Show Percentage displays the percentages of values that contribute to the segments in a pie chart.

CombineThreshold combines all values that are below a certain threshold. For example, if the range is 0 to 1, and you combine segments of less than 5%, the value combined is entered as 0.05 in the input.

Label provides a label to the combined segments.

!!! note In case of trouble in displaying Pie Chart data, try changing the query's function to PercentageofTimeAs.

Scalable Vector Graphics (SVG)

The SVG panel can display displayng metric-sensitive SVG images. SVG is useful for defining vector-based graphics for websites in XML format. All elements of the attributes in SVG files can be animated, and animation events can be triggered by incoming data.

Demo SVGs

Check out the SVG demonstration examples provided with the SVG panel by navigating to the bottom of the options page and clicking the corresponding button.

Use SVG Builder allows the use of the SVG Builder.

SVG Data opens a field into which you can input SVG data.

Events: allows you to write events in JavaScript so your code executes upon every refresh.

Add SVG repository facilitates users' custom SVG graphics by forking the original project and adding them to the assets folder. If your repository is of general concern, and your license allows sharing, you can add it to the panel plugin via a pull request.

SVG Data pastes your svg code here. You must include a viewbox and IDs for all relevant objects.

!!! note You cannot use the SVG Data editor together with the SVG Builder option checked!

onHandleMetric events execute upon every refresh; onHandleMetric(ctrl: MetricsPanelCtrl, svgnode: HTMLElement) ctrl passes a Dashboard_Designer MetricsPanelCtrl object that contains all data relevant to the current panel. You may want to use the ctrl.data array property to access the current measurement data.

svgnode passes the HTMLElement of the svg object on the panel. You can access the elements of the svg itself by using the integrated Snap Library. (http://snapsvg.io/)

Separator

Creates a blank panel you can use as a separator between dashboard panels.

ImageIt

Displays an image and overlays Insights Hub data on top of it.

Settings

Image URL is where you enter the URL of the image you want to display.

To display an image from an IoT file service, input the following address:

.

Size Coefficient scales the size of the image.

Query you can add query values on top of the image using the sensor feature. For example, on the image of a factory floorplan, you can display the values of assets in different locations directly over the asset image on the floorplan.

Metric selects a previously selected query and displays it on top of the image.

Postfix overrides the name of the query.

Unit adds a unit to the value.

Decimals sets the number of decimals to display.

Display Name adds the name you configure to the variable, before the prefix.

Size Coefficient scales the size of the value.

Position selects the position of the sensor according to its X- and Y-coordinates.

Link adds a link that you can click to navigate to; for example, the link could open a dashboard related to the asset.

Hover text specifies text to display when users hover over the value.

Appearance changes how the value is displayed on top of the image. You can also create color mappings that trigger when value thresholds are crossed.

Traffic Lights

Show status data in traffic lights.
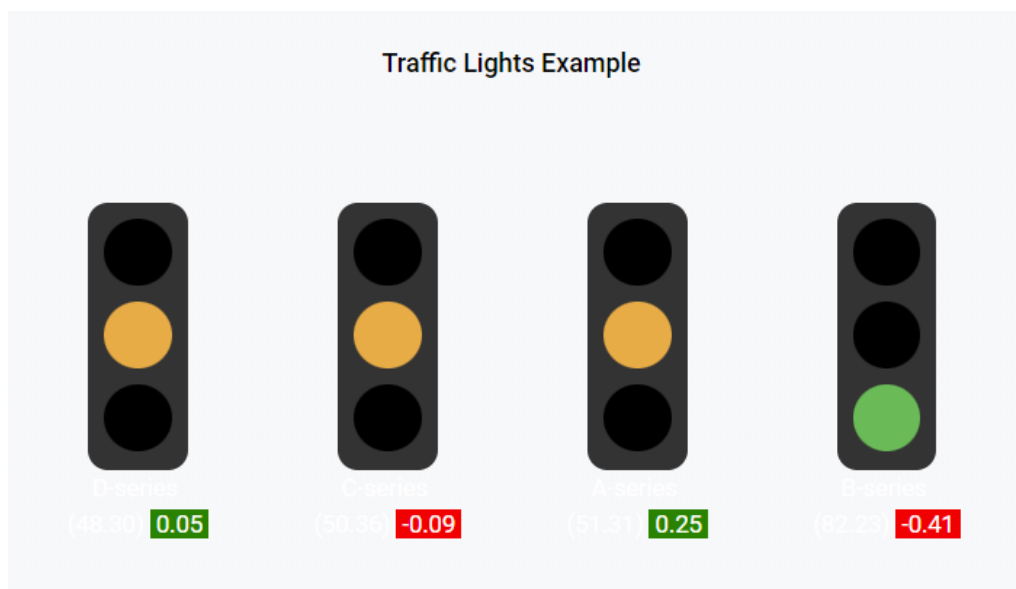
Traffic Light Options

Width limits the traffic light to a specified pixel width.

Font Size changes the font size displayed on the traffic light.

Font Color changes the font color displayed on the traffic light.

Show Value displays the value on the traffic light.

Unit adds a unit to the value.

Digits sets how many Digits are displayed.

Show Trend shows the trend of the data within the set time range.

Sort Lights changes the position of lights according to the values.

Render as link renders the traffic light as a link. Set up the URL in the Link section.

Design spreads traffic lights equidistant across the panel when more than one query is running.

Thresholds: Invert Scale Inverts the traffic light colors to display green at the top and red at the bottom.

Thresholds changes the threshold that activates the lights.