

2.0 Custom Dashboards for Beginners

Once you get the hang of the elements that are provided for you automatically, you may want to take the next step and start creating custom dashboards of your own.

FileWave supplied dashboards will only take you just so far. As soon as you want to know specific information about your environment and your deployments you are going to want to create your own dashboards with your own dashboard elements.

The contents for customization are broken into two parts. In the first (this section) we'll look at the components of dashboards and show you how you can build your own simple custom dashboards. In the next section, we'll get much more detailed and advanced.

- [Using Grafana for Data Aggregation](#)
- [2.1 Creating Your Own Dashboard](#)
- [2.2 Copying Widgets \(Panels\)](#)
- [2.2.1 Copied Panel is Blank](#)
- [2.3 Widget/Panel Elements](#)
- [2.4 Creating a New Panel \(existing data\)](#)

Using Grafana for Data Aggregation

What

Grafana has been included within FileWave for quite some time, but only with a recent update does Grafana have the ability to do data aggregation without using Prometheus. This article shows you how to use this feature.

When/Why

We'll use Data Aggregation whenever we want to look at an overview of data. For instance, if we want to understand how many devices are missing OS patches, we might create a report showing all devices, and their missing patches. To give a graphical representation of this, we would likely Group this data by patch name, and Count the number of devices missing each. This data is perfect for a visualization like a bar or pie chart.

How

To create an aggregated visualization, we need to start with a report in FileWave. In this instance, we'll create a report on the version of the FileWave client on Mac and Windows devices:

Name: FileWave Client VersionsMain Component: All Devices

☐ Include Archived Clients

CriteriaFields

All of these expressions must be true

One or more of these expressions must be true

☐ Not Operating System / OS Type is macOS

☐ Not Operating System / OS Type is Windows

☒ Not Desktop Device / FileWave Client Version is not null

Note that we included that the client versions is not null...this prevents having placeholders, etc in the data. Also note that we only included two fields here: FileWave Client Version (which we intend to Group By), and Device ID (which we intend to count). There aren't may devices in this system, so the data just looks like this:

Inventory Queries	
FileWave Client Versions (2) X	
Device ID	FileWave Client Version
0f5b4dcfdad12e8043827cf4bfc365de6da11c1a	15.3.0
8a88b8d0ecfe127fc2eb7dbd60c4858fd2203b32	15.1.0

Now we are ready for the "dashboard" part of the exercise, so go into your Dashboard from FileWave Anywhere and follow along with this short video:



Related Content

- [Dashboard \(Grafana\)](#)
- [Content Packs](#)

2.1 Creating Your Own Dashboard

What

The beauty of the FileWave dashboard isn't in what we give you...although we definitely give you some great stuff! The beauty of the solution is that you can make it what you want it to be using your own dashboards.

When/Why

Dashboard elements (or widgets as we will call them) can be combined to create dashboards of your own. You can copy these widgets from pre-existing FileWave Dashboards, or even create your own widgets directly.

How

In this example, we'll just look at creating the new empty dashboard we want, but do make sure and follow this up with the article below on copying widgets. Note that we create a new folder first, as we expect we'll create more than one dashboard eventually:



2.2 Copying Widgets (Panels)

What

If we have the power to create our own dashboards, then surely we must be able to add content to them...

When/Why

We are going to copy a widget (referred to as a panel in Grafana) whenever one already exists and we can leverage it without doing any extra work. For instance, watch below, as we "steal" content from various pre-existing dashboards for our own custom dashboard.

How

Taking pre-existing content from other dashboards is as easy a copy and paste (rinse and repeat):



If you find that your copied panel is blank, take a look at resolving that in this article: [Fixing Blank Copied Panel](#)

2.2.1 Copied Panel is Blank

What

In certain circumstances, when you copy/paste a panel into another dashboard, you may find that the new panel is blank.

When/Why

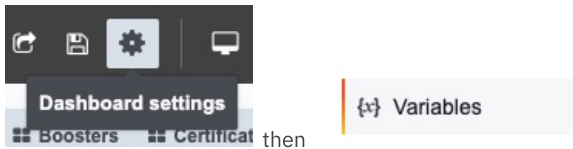
This will happen most often if the dashboard that you copied from has variables defined that the panel relies upon. For instance, if a panel relies on `$host` to be the address of your server, but the destination dashboard doesn't have that variable, then the panel will copy, but be blank.

How

Below you'll see the issue:



Fixing the issue requires replicating the variables in the destination dashboard. (Unfortunately there is no method for copy/paste, so they must be regenerated.). In any dashboard, you can see the variables in Dashboard Settings → Variables:



Replicate the variables in your own dashboard:

Variables > Edit

General

Name	host	Type	Query
Label	optional display name	Hide	

Query Options

Data source	FileWave Prom...	Refresh	On Dashboard Lo...
Query	label_values(up{job=~"postgres.*"},instance)		
Regex	/.*-(.*)-.*		
Sort	Disabled		

Selection Options

Multi-value	<input type="checkbox"/>
Include All option	<input checked="" type="checkbox"/>
Custom all value	.*

Value groups/tags (Experimental feature)

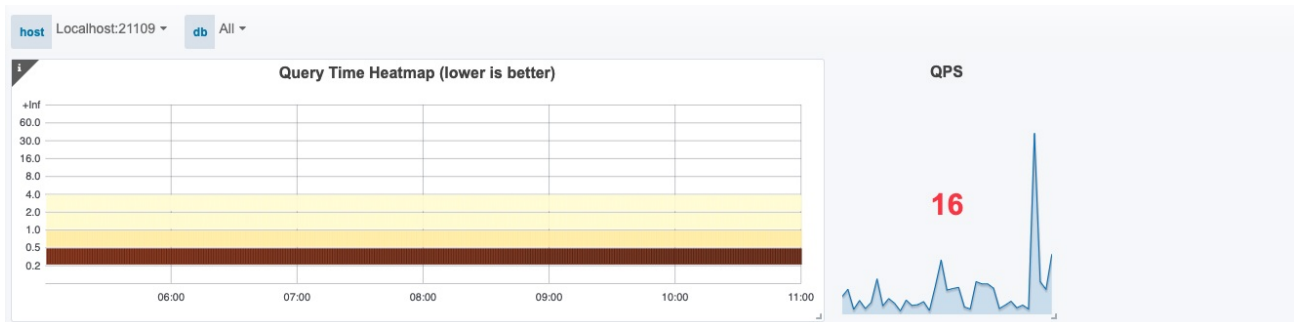
Enabled	<input type="checkbox"/>
---------	--------------------------

Preview of values

All	localhost:21109
-----	-----------------

Update

And, once you save those new variables, you should find your copied panel now works:



2.3 Widget/Panel Elements

What

All panels (or widgets) on the FileWave dashboard are comprised of the same basic elements. This article reviews those elements at a high level.

When/Why

If you are using pre-existing panels, you won't care too much about how they are built. However, as soon as you want to build your own panels, the building blocks become quite important.

How

All panels have the following three elements:

- The Query (Choosing what data you want to see in the panel)
- The Visualization (Choosing the visual representation of the data: gauge, bar chart, table...)
- General (Panel properties such as title, comment, links)

In the below you'll see examples of each of these elements:

The Query

Query

FileWave Tabular Datasource

All macOS Clients

Relative time

1h

Time shift

1h

The query you choose defines the data provided to the panel. In this case, an inventory query was selected.

The Visualization

Visualization

Table

Data

Table Transform

Table

Columns

Auto

Paging

Rows per page

100

Font size

100%

Column Styles

Options

Apply to columns named

Time

Column Header

Time

Render value as link

Type

Type

Date

Date Format

YYYY-MM-DD HH:mm:ss

Because we chose an inventory query as our data source, a table was our only possible visualization. In our more advanced example though, you'll see that we can do charts and graphs here of all sorts.

General

Title

macOS Clients

Description

Panel description, supports markdown & links

Transparent

☐

Repeating

Repeat

Note: You may need to change the variable selection to see this in action.

Panel links

We only specified a name in this case for this panel, but it is very useful to also set a link for the panel to tie directly into a report in the webadmin.

2.4 Creating a New Panel (existing data)

What

It is possible in the FileWave dashboard to add panels directly from an inventory query (report) with some limitations (no data aggregation).

When/Why

The data panels are simple enough to add as you'll see below, but the only representation available for them is a table format, which is the equivalent of the report/inventory query itself, but in the dashboard, so easier to share.

How

All Inventory Queries (reports) are available automatically when you choose FileWave Tabular Datasource for your panel's query. See a quick walkthrough below of creating a new table based panel to show all macOS clients:

