Tile Builder
Grid
Version 1.0

Revised 12/30/23

Table of Contents

Overview

Previous versions of Tile Builder have limited the user to two columns of data. The left-hand column for the device name and the right-hand column for the data. Tile Builder Grid allows the user to create tables up to 5 columns wide and place text and data anywhere they wish within that Grid. For the purposes of this document, I'm going to assume the reader is already familiar with using one of the other Tile Builder apps and knows how to perform the normal table editing functions.

Tile Builder Grid has two modes of operation.

Mode: Device Group

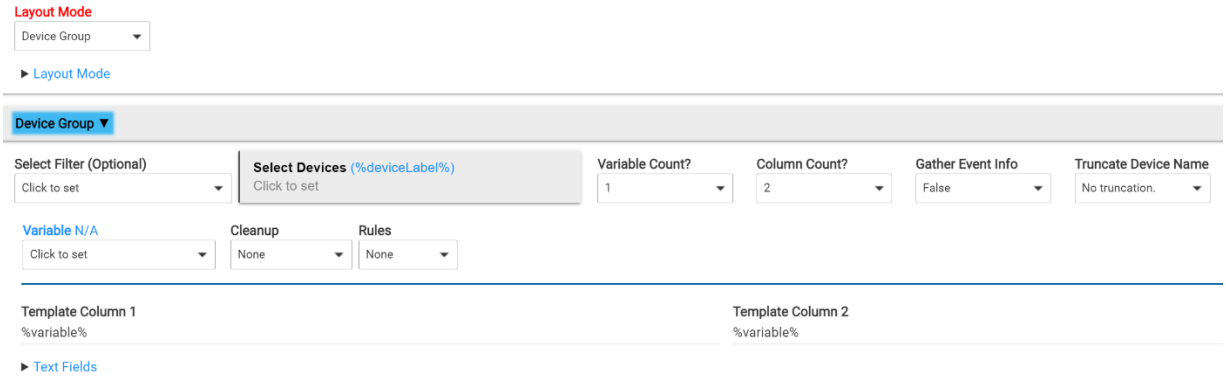
In this mode the user selects a group of devices they are interested in and fills out a one-line template for how the resulting data will look. That one line template is then applied to each device in the device group to create the final table. In this mode the information is always sorted by the device label.

Mode: Free Form

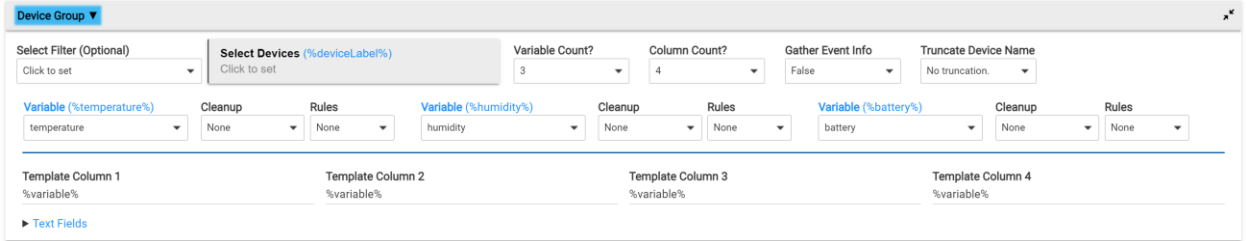
In this mode the user must create a template for the entire table. This mode is used for non-repeating data such as a weather tile, hub info tile or any kind of blended information. That template is then filled with the data for publication. In this mode the information layout always remains the same.

Layout: Device Group

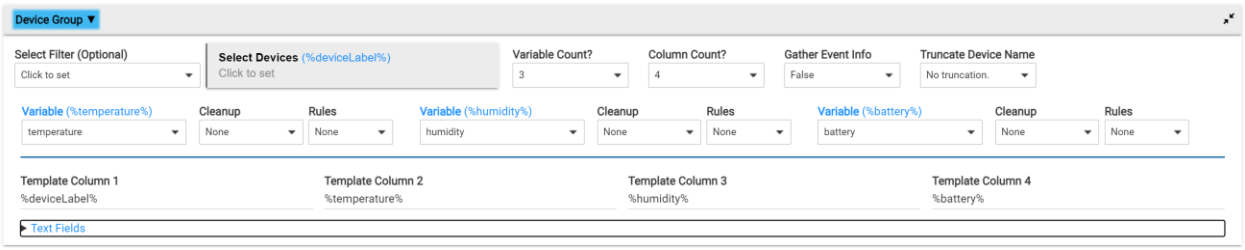
Let's say I want a Table that shows me Temperature, Humidity and Battery level for a range of devices. If I create a new Grid it will look like this when I launch it.



We will start with a simple case and put each piece of data in its own column. So, we will need 4 columns and 3 variables (temp, humidity, and battery). The screen would look like this.



Now we must place our variables within the template as shown below.

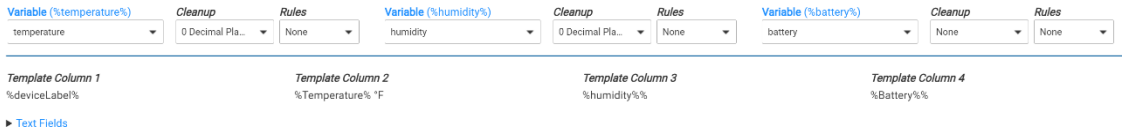


All we must do now is select the devices we want to display within the table. If we want, we can filter the list of devices by a specific capability using the **Select Filter (Optional)** control. Once I select my devices the table will look something like this.

Device	State	Other 1	Other 2
Attic Vent Fan & Sensor	37.6	82	N/A
Dining Room Thermostat	68.0	36	N/A
Living Room Sensor	70.36	38.2	100
Office Thermostat	68.49	N/A	66
OpenWeather	32	91	N/A
Temp Guage	19.7	37	21

10:13 AM

You can see that some fields have N/A indicating that the device does not have that attribute. The table looks O.K. but needs a little cleanup. If you have used Multi-Attribute Monitor, then the next few steps will be familiar. I'm going to shorten the device name, add units, and change the reported values to 0 decimal places like this:

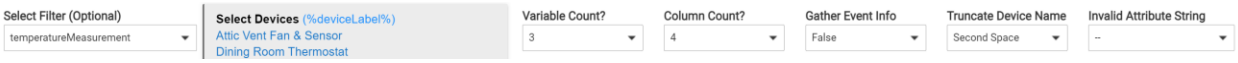


Here is the result of those changes.

Device	State	Other 1	Other 2
Attic Vent	38 °F	82%	N/A%
Dining Room	68 °F	36%	N/A%
Living Room	70 °F	38%	100%
Office Thermostat	68 °F	N/A%	66%
OpenWeather	32 °F	89%	N/A%
Temp Guage	20 °F	37%	21%

10:22 AM

Better, but still room for improvement. We need some better headers; shrink the table to recover space and I don't like the N/A%. I'm going to assume you know how to fix the first two items yourself. The third can be changed using the **Invalid Attribute String** field on the far right shown below (a late addition, not present in some other screen shots).



To finish it off I changed the Style to "Marooned" with overrides disabled and the result looks like this:

Device	Temp	Humidity	Battery
Attic Vent	40 °F	81%	--%
Dining Room	68 °F	36%	--%
Living Room	71 °F	38%	100%
Office Thermostat	67 °F	--%	66%
OpenWeather	33 °F	87%	--%
Temp Guage	20 °F	37%	21%

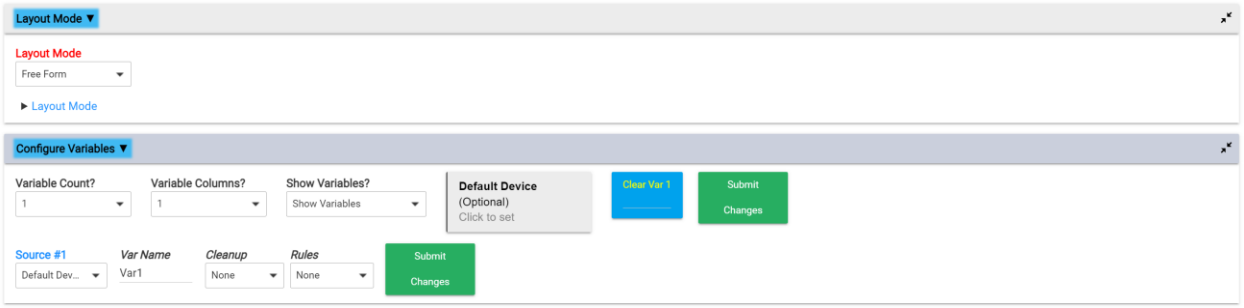
The size of this table is 853 bytes but reduces to 674 bytes with Aggressive scrubbing.

That is all you need to get started with a **Device Group,** but there is plenty more to learn in the following pages.

Layout: Free Form

Free Form mode is used to present a lot of unique pieces of data. Unlike a Device Group, the data does not repeat with each row.

To create a Free Form layout, we must first change the Layout mode as shown below.

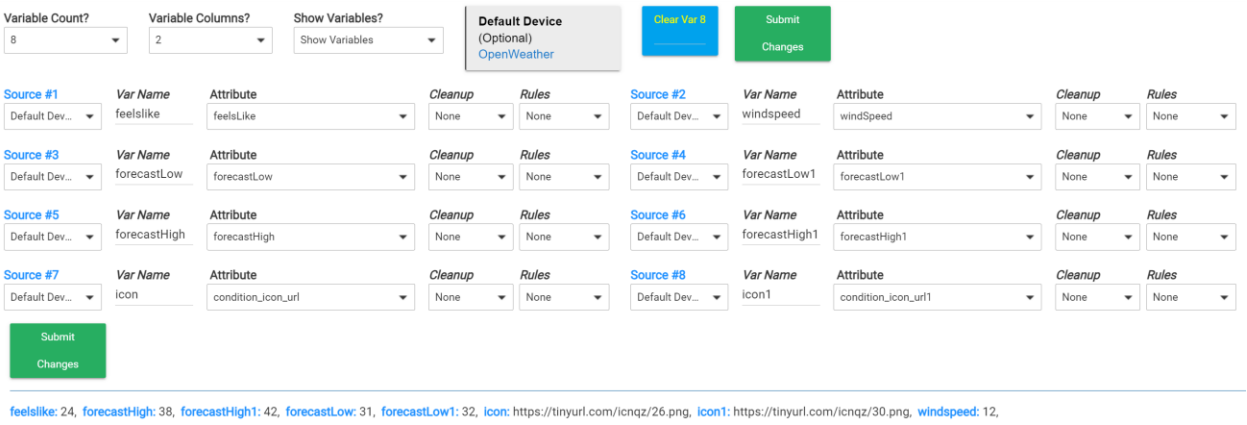


First, we are asked to configure the variables. Each variable is just a reference to an attribute and using the UI you will connect the variable to the attribute of interest. You can define up to 40 variables in addition to multiple built-in variables described later. For this example, I'm going to create a very simple Weather Tile using the Open Weather driver as the source.

The data I'm interested in is as follows. Today: current feel, wind, forecast high\low temp and forecast. Tomorrow: predicted high\low temp and forecast. That is a total of 8 variables.

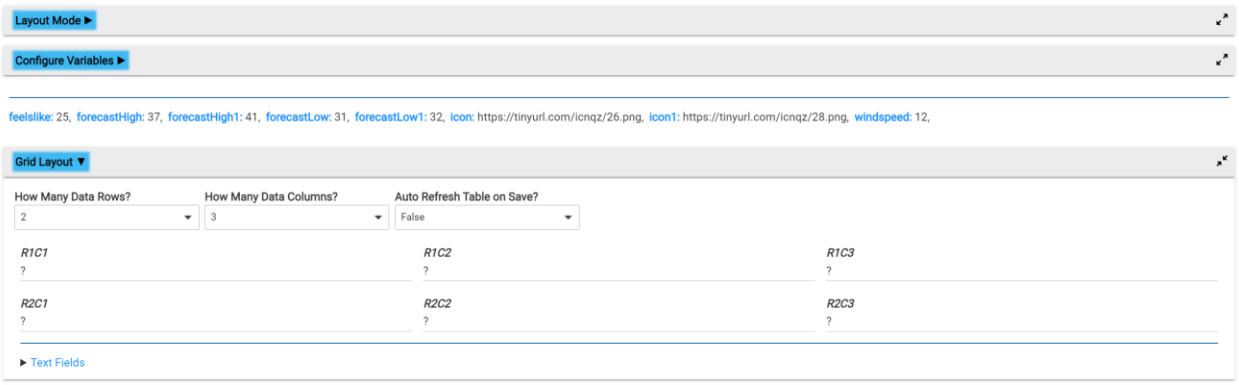
Because all this data is coming from my OpenWeather device I'm going to make that my **Default Device**. This simplifies the process of picking the device\attribute combination which can be tedious if you are selecting a lot of data. I have a widescreen monitor so I changed my **Variable Columns** to 2 and make it easier to view.

Here is my filled-out Variables section. Notice the variables and their values display at the bottom of the section as I proceed to fill it out. I have everything I need for this example so I can move on.



Note: Fields that are italicized do not automatically cause the Table to refresh when edited.

You can collapse the **Layout Mode** and **Configure Variables** sections to give yourself a little more room. Open the **Grid Layout** section and configure the **Data Rows** to be 2 and **Data Columns** to be 3. Your screen will look like this.

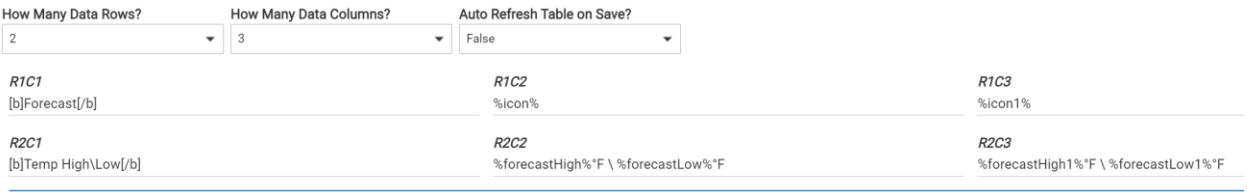


I'm going to use column 1 for the text labels, column 2 for today's data and column 3 for tomorrow's data. I can now place my variables like this.

Device	State	Other 1
Forecast	https://tinyurl.com/icnqz/26.png	https://tinyurl.com/icnqz/28.png
Temperature High\Low	37\ 31	41 \ 32

We have most of the raw information in there except for the Feel and Wind. We will add those later but let's dress this up a bit first including the headers.

Here we have added some units as well as some bold tags.

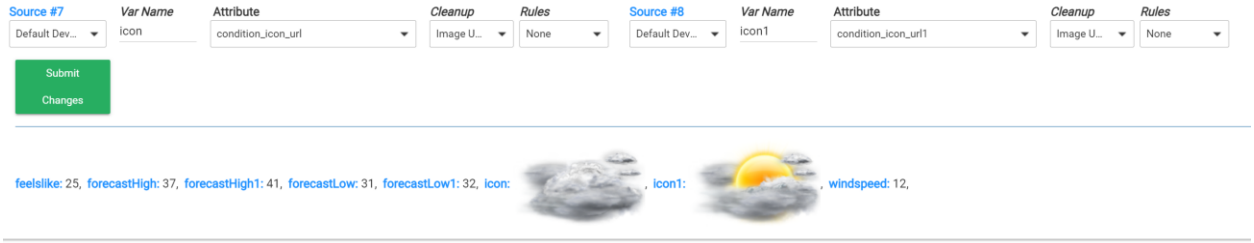


Added some header information like this:

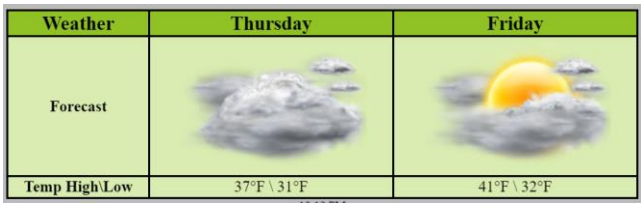
Now our table looks like this:

Weather	Thursday	Friday
Forecast	https://tinyurl.com/icnqz/26.png	https://tinyurl.com/icnqz/28.png
Temp High\Low	37°F \ 31°F	41°F \ 32°F

To get the actual weather icon to display as an image instead of a string we need to go back and apply a cleanup rule to force this change. In this case we tell it to treat the result as an image URL.

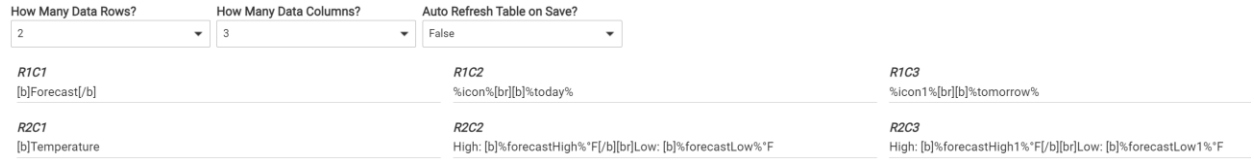


And the table looks like.

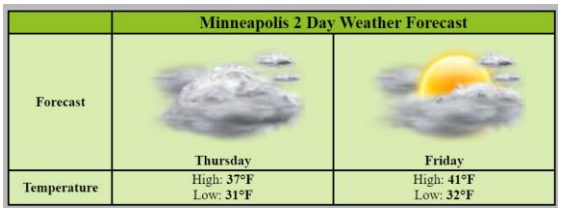


Weather	Thursday	Friday
Forecast		
Temp High\Low	37°F \ 31°F	41°F \ 32°F

Add a few more tags like this:



…..and merge the column 2 and 3 headers.



	Minneapolis 2 Day Weather Forecast	
Forecast	Thursday	Friday
Temperature	High: **37°F** Low: **31°F**	High: **41°F** Low: **32°F**

Note: The images presented using Open Weather are a fixed size and do not scale.

Almost there, just need to add the Feels Like temperature and the Wind Speed. I'm going to add them to the header like this.

This is the result.

Feel: 24°F	Minneapolis 2 Day Weather Forecast (14 m.p.h.)	
Forecast	 Thursday	 Friday
Temperature	High: **37°F** Low: **31°F**	High: **41°F** Low: **32°F**

Of course, this is just a simple exercise to demonstrate the basic principles. My actual weather tile is considerably more complex and looks like this:

Feel: 24°F	Minneapolis 3 Day Weather *No current weather alerts for this area*		Current: 33°F
Updated @ 12:17 PM on Thursday	Thursday  Overcast clouds	Friday  Broken clouds	Saturday  Scattered clouds
Low \ High	31°F \ 37°F	32°F \ 41°F	29°F \ 34°F
Wind	14 gusting to 14	North-Northwest (340°)	
Lux: 6,400 **UV:** 0.89	**Sunrise:** 07:50 **Sunset:** 16:39	**Humidity:** 83% **DewPoint:** 29%	**Rain Today:** 0.0 **Rain Tomorrow:** 0.0

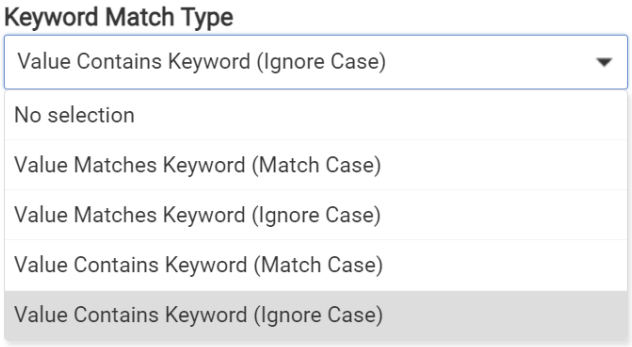
You now have all the basics you need to start creating sophisticated tables. However, there is still a lot to learn, and the next section covers more advanced topics.

Tile Builder Grid: Advanced Topics

I am assuming that you have already been using a Tile Builder module in Advanced mode and are already familiar with how to use the features under the Highlights tab. Namely, Keywords, Thresholds and Format Rules (MAM) so in this section I will only point out a few changes.

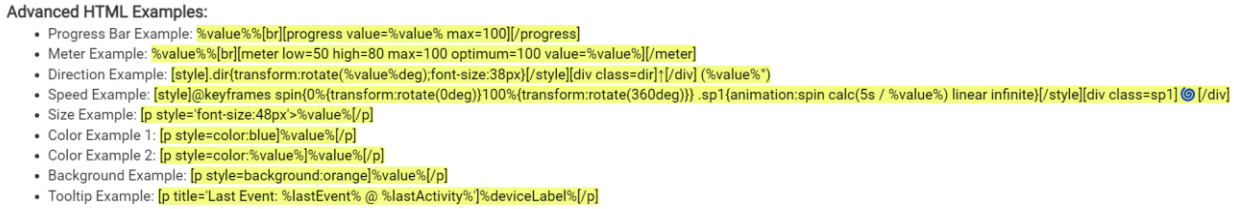
Highlights Tab

Keywords: You can now select the type of match you wish to test for with the options shown below.



Thresholds: No changes

Format Rules: These work the same as they do in Multi-Attribute Monitor but there are additional examples that can be found in the **Text Fields** help as shown below.



Replace Chars: This was introduced in Multi-Attribute Monitor and works the same way in Grid.

Built-In Variables

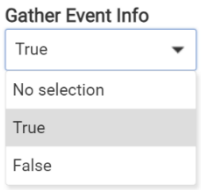
Grid has several built-in variables that you can place anywhere using the same %variableName% syntax. The following built-in variables are available and can be viewed under the **Text Fields** help. Variables are not case sensitive.

Built-In variables - Any Layout Mode

- %day% - Day of week in form: Fri
- %date% - Date in form: 22-12
- %date1% - Date in form: Dec-22
- %time% - Time in form: 23:35 PM
- %time1% - Time in form: 23:35
- %time2% - Time in form: 11:35 PM
- %today% - Current day as day of week in form: Friday
- %tomorrow% - Tomorrow as day of week in form: Saturday
- %dayAfterTomorrow% - Day after tomorrow as day of week in form: Sunday

Additional Built-In variables for Device Groups

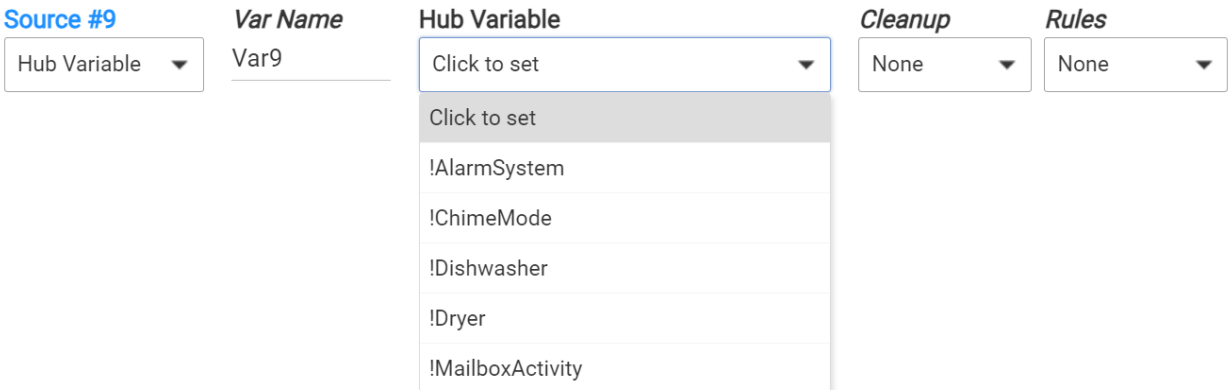
- %deviceName% - Name of the device.
- %deviceLabel% - Label of the device.
- %lastOn% - Last time 'switch' was turned on. N/A if not applicable or not available.
- %lastOff% - Last time 'switch' was turned off. N/A if not applicable or not available.
- %lastOpen% - Last time 'contact' was opened. N/A if not applicable or not available.
- %lastClosed% - Last time 'contact' was closed. N/A if not applicable or not available.
- %lastActivity% - Date and time of Last Activity on the device.
- %lastEvent% - Last event that occurred on the device and the corresponding value.



Note: For the lastOn, lastOff, lastOpen, lastClosed, lastActivity and lastEvent variables to be populated we must set **Gather Event Info** to True as shown above. Leave this set to **False** if you are not actively using any of these variables.

Hub Variables

In Tile Builder Grid you can now access Hub Variables directly (no device required) when using Free Form mode. Simply select Hub Variable as the source and select the hub variable of interest.



Open Weather Customizations

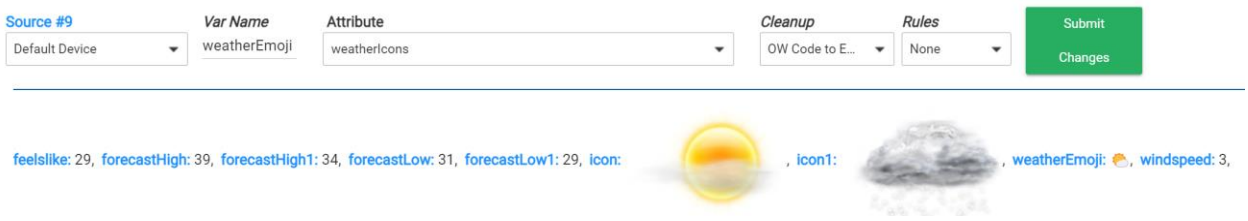
Open Weather seems to be the most prolific weather app used by the Hubitat community, so I added a couple of "**Cleanups**" related to Open Weather.

Load Image

The first one we used earlier which forced a string URL to load as an image. Technically this is not strictly Open Weather specific, but it is the only source for which I have tested compatibility.

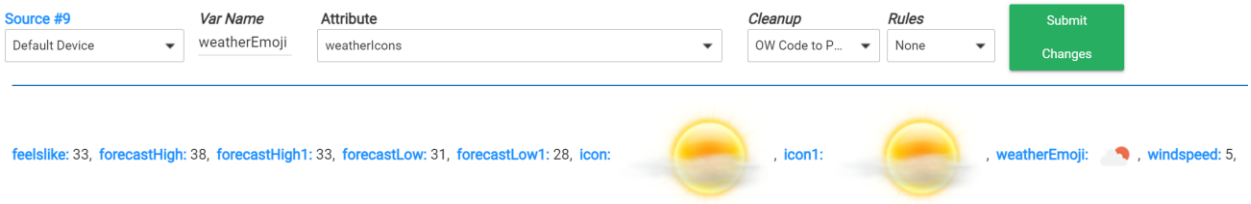
Use Weather Emoji (OW Code to Emoji)

Open Weather uses a code within the **weatherIcons** attribute that can be used to map the current weather to the closest available weather eMoji. This can be used where saving space, both in terms of footprint and byte size is a primary concern. See example below.



Use Open Weather Icon (OW Code to PNG)

This also uses the **weatherIcons** attribute that can be used to map the current weather to the designated Open Weather icon as shown below.



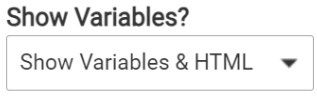
Weather Classes

There are a couple of classes listed under the **Text Fields** help that may be useful in displaying a weather tile and are not specific to Open Weather.

- Direction Example: [style].dir{transform:rotate(%value%deg);font-size:38px}[/style][div class=dir]↑[/div] (%value%°)
- Speed Example: [style]@keyframes spin{0%{transform:rotate(0deg)}100%{transform:rotate(360deg)}} .sp1{animation:spin calc(5s / %value%) linear infinite}[/style][div class=sp1]🌀[/div]

Viewing Variable HTML

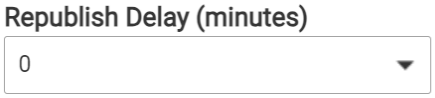
At times it is helpful to be able to view the underlying HTML that might be affecting a variable. You can do exactly that by changing the **Show Variables** control to **Show Variables & HTML** as shown.



In the example below I have added a rule to mark the feelsLike temperature variable in blue when the temperature falls below 35°F.

Publishing

Publishing your tile to the dashboard is just the same as in other modules with one small exception. I have added a setting called Republish Delay to throttle the number of publication events occur.

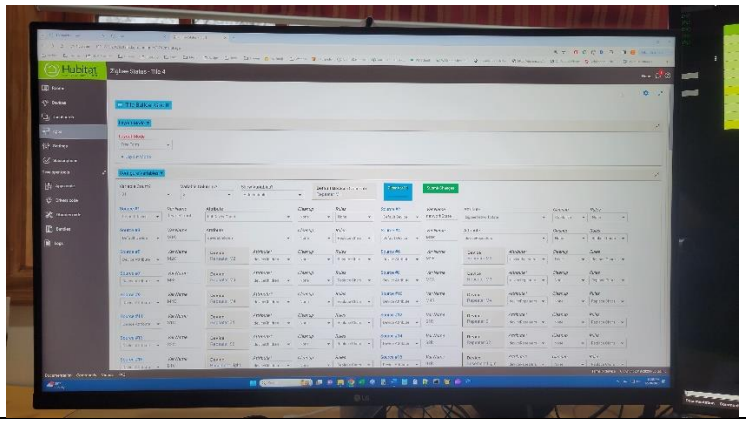


In my own experience I found I had some temperature sensors that were reporting changes very frequently and going from 67.91 to 67.92 which is overkill. The **Republish Delay** ensures that at least X minutes must pass since the last publication, before a Tile can be published again. The default is 0 which allows immediate publication subject to the **Event Timeout** settings.

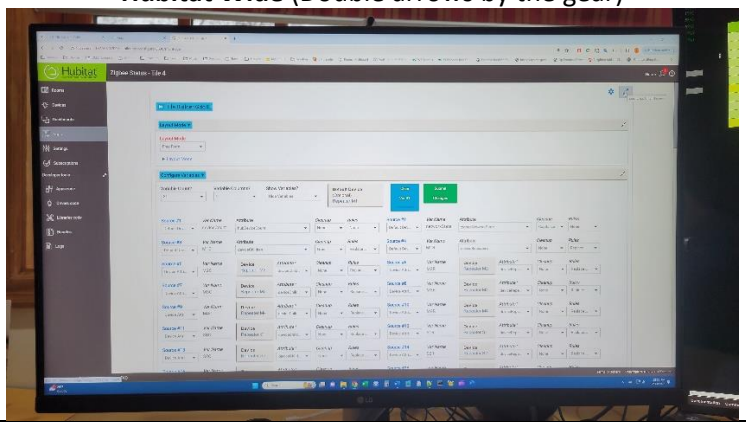
Full Screen Mode

Tile Builder Grid can have a lot of controls in use when creating a complex tile. To make the best use of the screen Tile Builder has always had collapsible sections using the headers. In Grid I have introduced a "Full-Screen" mode. This simply hides some of the Hubitat surrounding elements to leave more room for the important stuff as shown below.

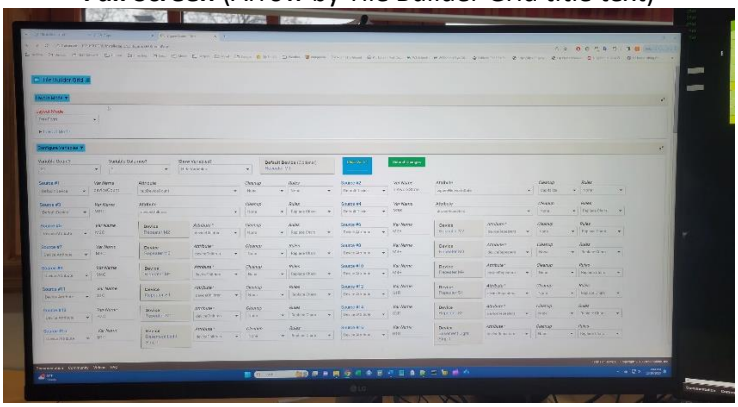
Normal



Hubitat Wide (Double arrows by the gear)



Full Screen (Arrow by Tile Builder Grid title text)



Overrides

Don’t forget that you can do all the same overrides that were available in other Tile Builder modules to make your tables even more fun!