

Tile Builder Grid Version 1.1

Revised 1/2/24

Table of Contents

Table of Contents	2
Overview	3
Layout: Device Group.....	3
Layout: Free Form	6
Tile Builder Grid: Advanced Topics	10
Highlights Tab.....	10
Keywords.....	10
Thresholds.....	10
Format Rules	10
Replace Chars.....	10
Built-In Variables.....	11
Hub Variables	11
Open Weather Customizations.....	12
Load Image.....	12
Use Weather Emoji (OW Code to Emoji)	12
Use Open Weather Icon (OW Code to PNG).....	12
Weather Classes.....	12
Viewing Variable HTML.....	13
Publishing	13
Full Screen Mode	14
Using Font-Awesome Icons.....	15
Hub Info Example.....	16
Adding a Title/Tooltip	17
Highlighting Updates.....	17

Tile Builder Grid – Version 1.1

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.

The screenshot shows the 'Layout Mode' interface for 'Device Group'. At the top, there's a 'Layout Mode' dropdown set to 'Device Group' and a '► Layout Mode' link. Below this is a 'Device Group' header with a dropdown arrow. The main configuration area includes: 'Select Filter (Optional)' with a 'Click to set' button; 'Select Devices (%deviceLabel%)' with a 'Click to set' button; 'Variable Count?' set to '1'; 'Column Count?' set to '2'; 'Gather Event Info' set to 'False'; and 'Truncate Device Name' set to 'No truncation.'. Below these are 'Variable N/A' with a 'Click to set' button, and 'Cleanup' and 'Rules' both set to 'None'. At the bottom, there are two template columns: 'Template Column 1' with the placeholder '%variable%' and 'Template Column 2' with the placeholder '%variable%'. A '► Text Fields' link is at the very bottom.

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.

Tile Builder Grid – Version 1.1

Device Group ▼

Select Filter (Optional)
Click to set ▼

Select Devices (%deviceLabel%)
Click to set

Variable Count?
3 ▼

Column Count?
4 ▼

Gather Event Info
False ▼

Truncate Device Name
No truncation. ▼

Variable (%temperature%)
temperature ▼

Cleanup
None ▼

Rules
None ▼

Variable (%humidity%)
humidity ▼

Cleanup
None ▼

Rules
None ▼

Variable (%battery%)
battery ▼

Cleanup
None ▼

Rules
None ▼

Template Column 1
%variable%

Template Column 2
%variable%

Template Column 3
%variable%

Template Column 4
%variable%

▶ Text Fields

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

Device Group ▼

Select Filter (Optional)
Click to set ▼

Select Devices (%deviceLabel%)
Click to set

Variable Count?
3 ▼

Column Count?
4 ▼

Gather Event Info
False ▼

Truncate Device Name
No truncation. ▼

Variable (%temperature%)
temperature ▼

Cleanup
None ▼

Rules
None ▼

Variable (%humidity%)
humidity ▼

Cleanup
None ▼

Rules
None ▼

Variable (%battery%)
battery ▼

Cleanup
None ▼

Rules
None ▼

Template Column 1
%deviceLabel%

Template Column 2
%temperature%

Template Column 3
%humidity%

Template Column 4
%battery%

▶ Text Fields

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:

Variable (%temperature%)
temperature ▼

Cleanup
0 Decimal Pla... ▼

Rules
None ▼

Variable (%humidity%)
humidity ▼

Cleanup
0 Decimal Pla... ▼

Rules
None ▼

Variable (%battery%)
battery ▼

Cleanup
None ▼

Rules
None ▼

Template Column 1
%deviceLabel%

Template Column 2
%Temperature% °F

Template Column 3
%humidity%%

Template Column 4
%Battery%%

▶ Text Fields

Here is the result of those changes.

Tile Builder Grid – Version 1.1

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).

Select Filter (Optional)
temperatureMeasurement

Select Devices (%deviceLabel%)
Attic Vent Fan & Sensor
Dining Room Thermostat

Variable Count?
3

Column Count?
4

Gather Event Info
False

Truncate Device Name
Second Space

Invalid Attribute String
--

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.

Device 1
Bedroom Sensor

Item Name
Bedroom Temp

Prepend
Click to set

Attribute *
temperature

Append
Click to set

Cleanup
None

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

Tile Builder Grid – Version 1.1

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.

The screenshot shows the 'Layout Mode' dropdown set to 'Free Form'. Below it, the 'Configure Variables' section is visible, showing settings for 'Variable Count?' (1), 'Variable Columns?' (1), 'Show Variables?' (Show Variables), 'Default Device (Optional)' (Click to set), 'Source #1' (Default Dev...), 'Var Name' (Var1), 'Cleanup' (None), and 'Rules' (None). There are 'Clear Var 1' and 'Submit Changes' buttons.

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.

The screenshot shows the 'Variables' section filled out with 8 variables. The settings at the top are: 'Variable Count?' (8), 'Variable Columns?' (2), 'Show Variables?' (Show Variables), 'Default Device (Optional)' (OpenWeather), 'Clear Var 8', and 'Submit Changes'. The variables are listed in two columns:

Source	Var Name	Attribute	Cleanup	Rules
Source #1	Default Dev...	feelslike	feelsLike	None
Source #2	Default Dev...	windspeed	windSpeed	None
Source #3	Default Dev...	forecastLow	forecastLow	None
Source #4	Default Dev...	forecastLow1	forecastLow1	None
Source #5	Default Dev...	forecastHigh	forecastHigh	None
Source #6	Default Dev...	forecastHigh1	forecastHigh1	None
Source #7	Default Dev...	icon	condition_icon_url	None
Source #8	Default Dev...	icon1	condition_icon_url1	None

At the bottom, the values are displayed: feelslike: 24, forecastHigh: 38, forecastHigh1: 42, forecastLow: 31, forecastLow1: 32, icon: https://tinyurl.com/icnqz/26.png, icon1: https://tinyurl.com/icnqz/30.png, windspeed: 12.

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

Tile Builder Grid – Version 1.1

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.

Layout Mode ▶

Configure Variables ▶

feelslike: 25, forecastHigh: 37, forecastHigh1: 41, forecastLow: 31, forecastLow1: 32, icon: https://tinyurl.com/icnqz/26.png, icon1: https://tinyurl.com/icnqz/28.png, windspeed: 12,

Grid Layout ▼

How Many Data Rows?
2

How Many Data Columns?
3

Auto Refresh Table on Save?
False

R1C1 ?	R1C2 ?	R1C3 ?
R2C1 ?	R2C2 ?	R2C3 ?

▶ Text Fields

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.

How Many Data Rows?
2

How Many Data Columns?
3

Auto Refresh Table on Save?
False

R1C1 Forecast	R1C2 %icon%	R1C3 %icon1%
R2C1 Temperature High\Low	R2C2 %forecastHigh% \ %forecastLow%	R2C3 %forecastHigh1% \ %forecastLow1%

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.

How Many Data Rows?
2

How Many Data Columns?
3

Auto Refresh Table on Save?
False

R1C1 [b]Forecast[/b]	R1C2 %icon%	R1C3 %icon1%
R2C1 [b]Temp High\Low[/b]	R2C2 %forecastHigh%"F \ %forecastLow%"F	R2C3 %forecastHigh1%"F \ %forecastLow1%"F

Added some header information like this:

Heading 1
Weather

Heading 2
%today%

Heading 3
%tomorrow%

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

Tile Builder Grid – Version 1.1



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.

Source #7	Var Name	Attribute	Cleanup	Rules	Source #8	Var Name	Attribute	Cleanup	Rules
Default Dev...	icon	condition_icon_url	Image U...	None	Default Dev...	icon1	condition_icon_url1	Image U...	None

Submit
Changes

feelslike: 25, forecastHigh: 37, forecastHigh1: 41, forecastLow: 31, forecastLow1: 32, icon: , icon1: , windspeed: 12,

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:

How Many Data Rows?	How Many Data Columns?	Auto Refresh Table on Save?
2	3	False

R1C1 [b]Forecast[/b]	R1C2 %icon%[br][b]%today%	R1C3 %icon1%[br][b]%tomorrow%
R2C1 [b]Temperature	R2C2 High: [b]%forecastHigh%°F[/b][br]Low: [b]%forecastLow%°F	R2C3 High: [b]%forecastHigh1%°F[/b][br]Low: [b]%forecastLow1%°F

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

	Minneapolis 2 Day Weather Forecast	
Forecast		
Temperature	Thursday High: 37°F Low: 31°F	Friday 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.




Heading 1	Heading 2
Feel: %feelsLike%°F	recast (%windspeed% m.p.h.)

Tile Builder Grid – Version 1.1

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 <i>No current weather alerts for this area</i>		Current: 33°F
Updated @ 12:17 PM on Thursday	Thursday	Friday	Saturday
	 Overcast clouds	 Broken clouds	 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.

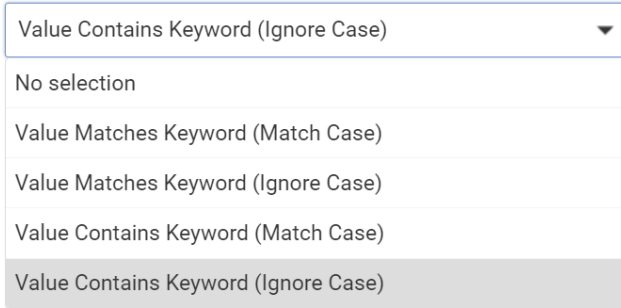
Tile Builder Grid: Advanced Topics

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.

Keyword Match Type



Value Contains Keyword (Ignore Case) ▼

No selection

Value Matches Keyword (Match Case)

Value Matches Keyword (Ignore Case)

Value Contains Keyword (Match Case)

Value Contains Keyword (Ignore Case)

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.

Advanced HTML Examples:

- Progress Bar Example: `%value%
[progress value=%value% max=100]/progress]`
- Meter Example: `%value%
[meter low=50 high=80 max=100 optimum=100 value=%value%]/meter]`
- 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]`
- Size Example: `[p style=font-size:48px>%value%]/p]`
- Color Example 1: `[p style=color:blue]%value%]/p]`
- Color Example 2: `[p style=color:%value%]%value%]/p]`
- Background Example: `[p style=background:orange]%value%]/p]`
- Tooltip Example: `[p title=Last Event: %lastEvent% @ %lastActivity%]%deviceLabel%]/p]`

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

Tile Builder Grid – Version 1.1

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.

Gather Event Info

True	▼
No selection	
True	
False	

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.

Source #9	Var Name	Hub Variable	Cleanup	Rules
Hub Variable ▼	Var9	Click to set ▼ Click to set !AlarmSystem !ChimeMode !Dishwasher !Dryer !MailboxActivity	None ▼	None ▼

Tile Builder Grid – Version 1.1

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.



Source #9	Var Name	Attribute	Cleanup	Rules	Submit
Default Device	weatherEmoji	weatherIcons	OW Code to E...	None	Changes

feelslike: 29, forecastHigh: 39, forecastHigh1: 34, forecastLow: 31, forecastLow1: 29, icon:  , icon1:  , weatherEmoji: 🌤️ , windspeed: 3,

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.

Source #9	Var Name	Attribute	Cleanup	Rules	Submit
Default Device	weatherEmoji	weatherIcons	OW Code to P...	None	Changes

feelslike: 33, forecastHigh: 38, forecastHigh1: 33, forecastLow: 31, forecastLow1: 28, icon:  , icon1:  , weatherEmoji: 🌤️ , windspeed: 5,

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:1[/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.

Show Variables?

Show Variables & HTML ▼

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

```
feelslike: 33 ([hq6]33/[hq6]), forecastHigh: 38 (38), forecastHigh1: 33 (33), forecastLow: 31 (31), forecastLow1: 28 (28), icon:  ([img src=https://tinyurl.com/icnqz/34.png]), icon1:  ([img src=https://tinyurl.com/icnqz/34.png]), weatherEmoji: 🌤 ([img src=https://openweathermap.org/img/wn/02d.png]), windspeed: 5 (5),
```

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.

Republish Delay (minutes)

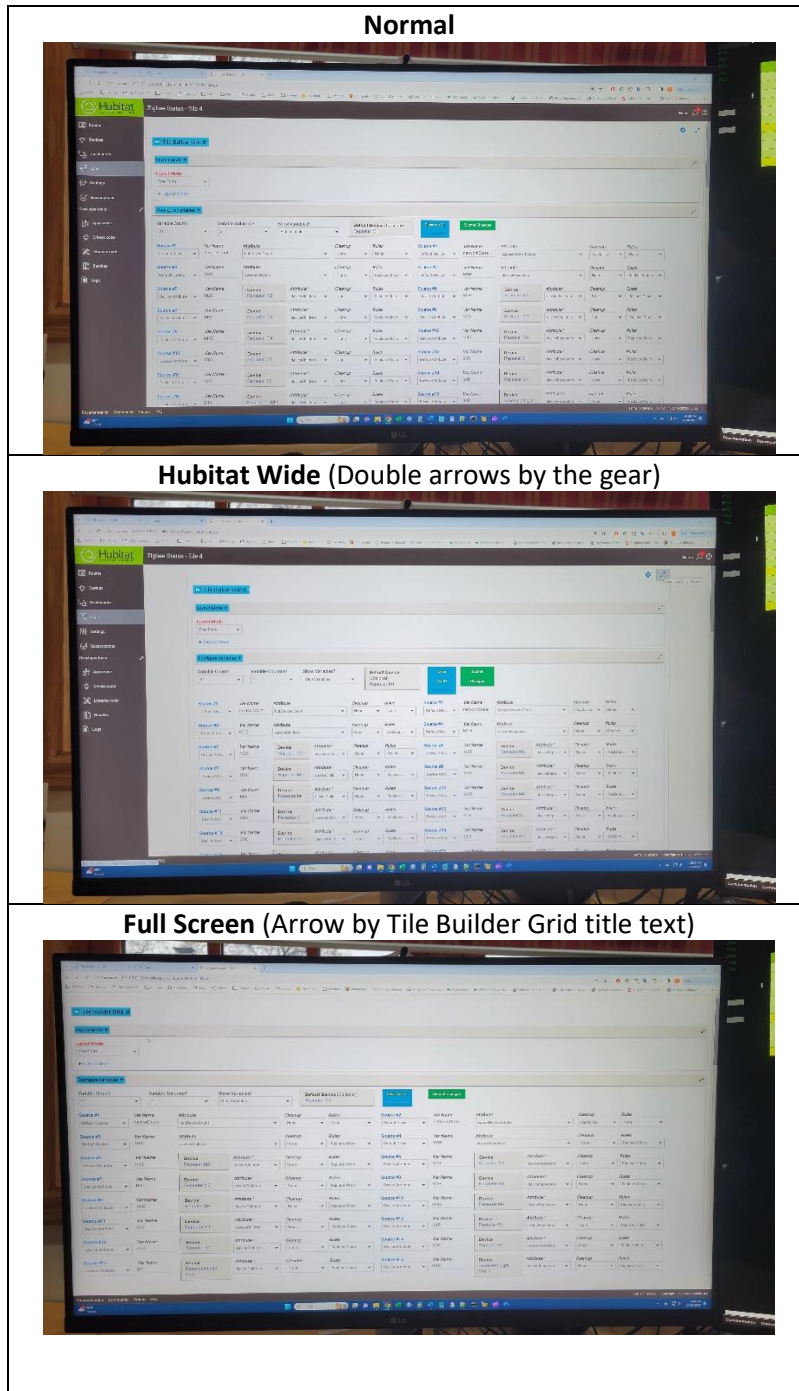
0 ▼

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.

Tile Builder Grid – Version 1.1

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.



Using Font-Awesome Icons

Using Overrides you can load alternate fonts and Icons if desired. Go to the Advanced Tab and click on Enable Overrides:

- 1) Add the following text to your Overrides area: **#head#=[link rel=stylesheet href='https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.0/css/all.css']**
- 2) Using a URL shortener I get this <http://tinyurl.com/5956hskv> so I could use: **#head#=[link rel=stylesheet href='http://tinyurl.com/5956hskv']** which saves some space.

If you have other commands in Overrides already you will need to separate them with the vertical bar symbol like this command1 | command 2 | command3

You can now place a Font-Awesome Icon using the following syntax in any text field: **[i class='fa-solid fa-temperature-full']** Doing so will render this icon  in whatever your selected Font Size is.

You can find a searchable list of free Font-Awesome icons at this URL:

<https://fontawesome.com/search?o=r&m=free>

When you click on an Icon you will get a popup like this:



Clicking on the code snippet will place it in the copy\paste buffer and you can paste it into any Tile Builder text field like this:

```
R5C1
<i class="fa-solid fa-house"></i>
```

You are not quite finished! You must change any <> to [] and any " to ' so the final result looks like this.

```
R5C1
[i class='fa-solid fa-house'][/i]
```

Note: Providing a closing tag is optional if it is the last item in the field so the [/i] could be omitted.

Tile Builder Grid – Version 1.1

Hub Info Example

Here is how I build my Hub Info Tile.

Configure Variables (Partial list)

Configure Variables

Variable Count?
16

Variable Columns?
1

Show Variables?
Show Variables

Default Device
(Optional)
Hub Info

Clear
Var 16

Submit
Changes

Source #	Var Name	Attribute	Cleanup	Rules
Source #1	Default Device	dbSize	0 Decimal Pla...	None
Source #2	Default Device	lastRestart	lastHubRestartFormatted	None
Source #3	Default Device	hubUpdateSta	hubUpdateStatus	All Keywo...
Source #4	Default Device	cpu5	cpuPct	None
Source #5	Default Device	cpu15	cpu15Pct	None
Source #6	Default Device	zigbeeStatus	zigbeeStatus	Capitalize
Source #7	Default Device	zWaveStatus	zwaveStatus	Capitalize
Source #8	Default Device	connectType	connectType	None
Source #9	Default Device	freeMem	freeMemory	Commas

Grid Layout

Grid Layout

How Many Data
Rows?
9

How Many Data
Columns?
2

Auto Refresh Table
on Save?
True

R1C1 [i class='fa-solid fa-temperature-full']	R1C2 %hubTemp%°F
R2C1 [i class='fa-solid fa-memory']	R2C2 [z title='5 / 15 min averages']%freemem% / %freemem15%
R3C1 [i class='fa-solid fa-microchip']	R3C2 [z title='5 / 15 min averages']%cpu5% / %cpu15%
R4C1 [i class='fa-solid fa-power-off']	R4C2 %lastRestart%
R5C1 [i class='fa-solid fa-network-wired']	R5C2 %connectType%
R6C1 [i class='fa-solid fa-database']	R6C2 %dbSize% Mb
R7C1 [i class='fa-solid fa-code']	R7C2 %version%
R8C1 [i class='fa-solid fa-cloud-arrow-down']	R8C2 %hubUpdateStatus%
R9C1 [i class='fa-solid fa-hockey-puck']	R9C2 %hubModel%

Result

Home Hub @ 10:50 AM

104°F

509,908 / 514,093

2.0% / 2.08%

01/01/2024
@ 04:14PM

Ethernet

6 Mb

2.3.6.146

Update Available

C-5

Alerts: platformUpdateAvailable

Tile Builder Grid

Page 16

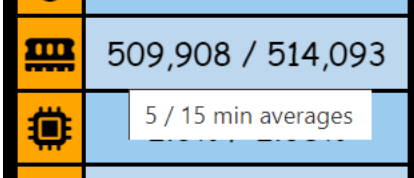
Section: 3

Tile Builder Grid – Version 1.1

To add the orange background to the first column and get the Icons the size I wanted I added this to my overrides: `#class1#=#td:nth-child(1) {font-size:20px !important;background:orange !important}`

This says for the 1st column make the font size 20 pixels with an Orange background. The !important tells HTML\CSS that this setting should override all competing settings.

Adding a Title/Tooltip

<p>The <code>[z title='5 / 15 min averages']%freemem% / %freemem15%</code> uses the HTML Title command to add a ToolTip to the text which I used for fields that have two numbers to identify the significance of each. The letter z is just an assigned class name and has no relevance, other than being very short.</p>	
<p>Overrides now looks like this:</p> <p>Settings Overrides ↔</p> <pre>#head#=[link rel=stylesheet href='http://tinyurl.com/5956hskv'] #class1#=#td:nth-child(1) {font-size:20px !important;background:orange !important}</pre>	

Highlighting Updates

Whenever there is an update, it will be shown in green text. This is done using a **Keyword** which can be found under the **Highlights** tab.

<p>Assign Keyword testing to the Variable.</p>	<p>Source #3 Var Name Attribute Cleanup Rules</p> <p>Default Dev... hubUpdateSta... hubUpdateStatus None All Keywo...</p>
<p>If the result contains “Update Available”, make it green.</p>	<p>Highlights</p> <p>Show Keywords ▼ How Many Keywords? 1</p> <p>Keyword Match Type Enter Keyword #1 Replacement Text #1 Highlight 1 Color (#008000) Highlight 1 Text Scale</p> <p>Value Matches Keyword (Ignore Case) Update Available [b]value[/b] 100</p>

It may seem a little tricky at first, but you will get used to it pretty quickly.