

Tile Builder Thermostat Help

Revised 6/18/24

Tile Builder Thermostat

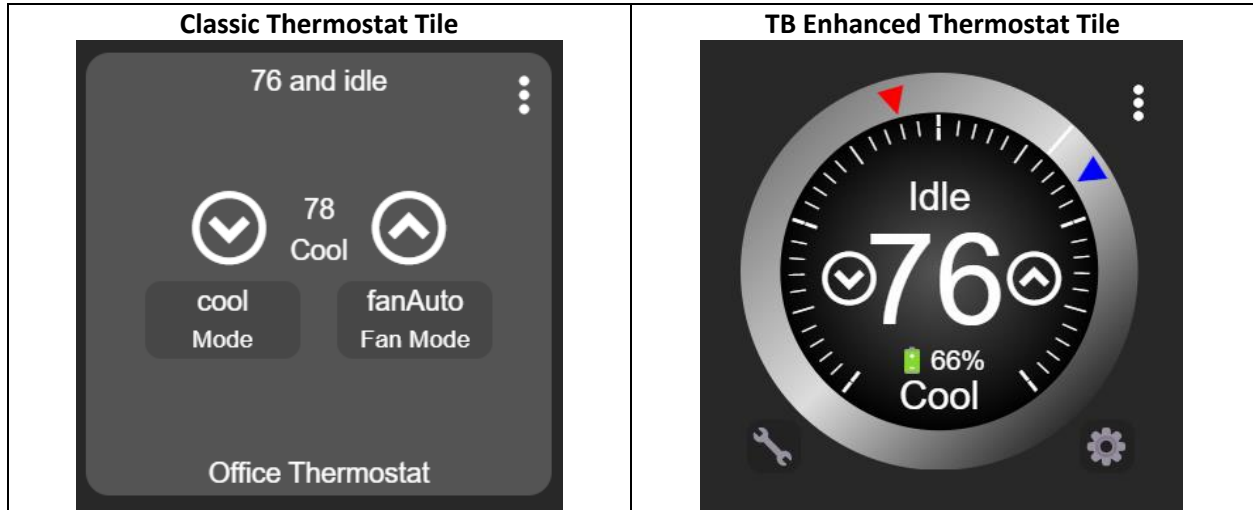
Table of Contents

Table of Contents	2
Introduction	3
Requirements.....	3
Creating a Tile Builder Thermostat	4
Understanding the Display.....	7
Customizing Your Thermostat.....	7
Testing Thermostat Modes	9
Managing Tile Size	10
Troubleshooting.....	11
Slow Response Times	11
Missing Controls.....	12
Tweaking Publishing Performance.....	12

Tile Builder Thermostat

Introduction

The classic Hubitat Dashboard is functional and customizable, but rather dated in appearance. Tile Builder Thermostat allows you to add an attractive face to the classic Thermostat tile and maintain all of the same functionality.



Requirements

You must have the Tile Builder Parent version 1.5.1 or later installed. Thermostat is an optional module and you must choose to **Modify** the Tile Builder installation to have the code added as shown below.

Modify a Package

Items below that are checked are currently installed. Those that are not checked are currently **not** installed.

Select the apps to install/uninstall

Tile Builder - Activity Monitor, Tile Builder - Attribute Monitor, Tile Builder - Multi Attribute Monitor, Tile Builder - Rooms, Tile Builder - Grid

- ☒ Tile Builder - Activity Monitor
- ☒ Tile Builder - Attribute Monitor
- ☒ Tile Builder - Multi Attribute Monitor
- ☒ Tile Builder - Rooms
- ☒ Tile Builder - Grid
- ☐ Tile Builder - Thermostat

Tile Builder Thermostat

Creating a Tile Builder Thermostat

To create a new Thermostat launch Tile Builder and select **Add New Thermostat**

Add New Thermostat

Manage Tiles ▶

The main screen will launch and look like this.

Select Thermostat ▼

Choose Thermostat

Click to set

Design Thermostat ▼

Refresh Thermostat

☐ Customize Thermostat

Choose a Thermostat in the **Select Thermostat** section at the top of the page.

Simulate Thermostat Mode

Use Thermostat ▼

Current HTML size is: **11** bytes. Maximum size for dashboard tiles is 1,024 bytes.

Important: You must copy the CSS from the **Tile Builder Dashboard CSS** tab to your **Habitat Dashboard CSS** screen for the Thermostat to work interactively.

Publish Thermostat ▼

Here you will configure where the table will be stored. It will be refreshed at the frequency you specify.

Tile Attribute to store the table? *

Click to set ▼

Name this Tile*

Click to set

For Reference Only: Tiles in Use

Click to set ▼

Event Timeout (millis)

250 ▼

▶ Publishing Controls

Publish and Subscribe

Delete Subscription

Tile Builder Thermostat

The configuration process is quite straightforward.

1) Select a Thermostat

Select Thermostat ▼

Choose Thermostat
Office Thermostat

2) Publish the Thermostat

Publish Thermostat ▼

Here you will configure where the table will be stored. It will be refreshed at the frequency you specify.

Tile Attribute to store the table? *

Name this Tile*

For Refr

tile5 ▼

Office Thermostat - Tile 5

Click to

► Publishing Controls

Publish and Subscribe

3) Copy the CSS from the **Dashboard CSS** tab which becomes visible when you click enable the **Customize Thermostat** control.

General	Display	Heating	Cooling	Idle	Other Modes	Dashboard CSS
---------	---------	---------	---------	------	-------------	---------------

Tile Builder CSS: This is the CSS used by Tile Builder Thermostat. You must copy these lines to your Dashboard CSS in order for the Dashboard to work correctly. Copy them exactly as shown here, including the comment lines and then edit to your needs. Using the Condense CSS option removes most comments.

Condense CSS
True ▼

```
/* This CSS generated by Tile Builder Thermostat version: v1.0.0 (6/17/24 @ 08:24 PM) */
:root {--BW:0px;}
.tile-title {visibility: hidden; display: none;}
[class*=attribute][background-color:rgba(128,128,128,0); outline:var(--BW) solid white]
[class*=thermostat][background-color:rgba(128,128,128,0) !important; outline:var(--BW) dashed purple; z-index:99 !important]
.thermostat div.self-start {color:rgba(128,128,128,0); outline:var(--BW) dotted red;}
[class*=he-circle-up][position:absolute; outline:var(--BW) dotted orange; font-size:25px !important; margin-left: 15px; margin-top:8px; padding:0px]
[class*=he-circle-down][position:absolute; outline:var(--BW) dotted orange; font-size:25px !important; margin-left:-46px; margin-top:8px; padding:0px]
.thermostat div.inline-block {color:rgba(128,128,128,0); outline:var(--BW) dashed yellow;}
.thermostat div.w-full.my-1 {color:rgba(128,128,128,0); outline:var(--BW) dotted blue; position:absolute !important; left:-60px; width:320px; top:161px}
.thermostat div.w-full.my-1>div.inline-block {color:rgba(128,128,128,0); width:25px; height:25px; outline:var(--BW) dotted green; display: inline-block; vertical-align: middle; margin-left:53px; margin-right:60px;z-index:100}
.thermostat div.absolute.bottom-0.text-center.w-full {color:rgba(128,128,128,0); outline:var(--BW) dotted violet; z-index:-1}
```

4) Close the Tile Builder Thermostat screen by clicking Done.

5) Authorize the Dashboard to use the Thermostat and the Tile Builder Storage Device.

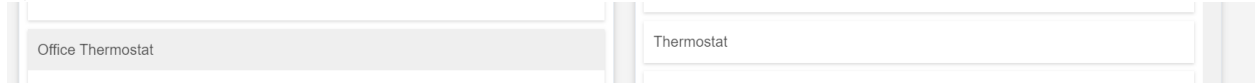
Tile Builder Thermostat

6) Add the CSS to the Dashboard by opening the Dashboard, selecting the **Gear Icon**, click on **Advanced**, then **CSS** and paste the CSS you copied from Tile Builder Thermostat into here. It should look like this.

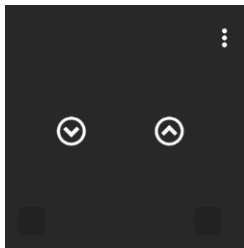
```
Layout Devices CSS Debug
Grid Options Templates Advanced

/* This CSS generated by Tile Builder Thermostat version: v1.0.0 (6/17/24 @ 08:24 PM) */
:root {--Bk:0px;}
.tile-title {visibility: hidden; display: none;}
[class*=attribute][background-color:rgba(128,128,0); outline:var(--Bk) solid white]
[class*=thermostat][background-color:rgba(128,128,0) !important; outline:var(--Bk) dashed purple; z-index:99 !important]
.thermostat div.self-start {color:rgba(128,128,0); outline:var(--Bk) dotted red;}
[class*=he-circle-up][position:absolute; outline:var(--Bk) dotted orange; font-size:25px !important; margin-left: 15px; margin-top:8px; padding:0px]
[class*=he-circle-down][position:absolute; outline:var(--Bk) dotted orange; font-size:25px !important; margin-left:-46px; margin-top:8px; padding:0px]
.thermostat div.inline-block {color:rgba(128,128,0); outline:var(--Bk) dashed yellow;}
.thermostat div.w-full.my-1 {color:rgba(128,128,0); outline:var(--Bk) dotted blue; position:absolute !important; left:-60px; width:120px; top:161px}
.thermostat div.w-full.my-1 div.inline-block {color:rgba(128,128,0); width:25px; height:25px; outline:var(--Bk) dotted green; display: inline-block; vertical-align: middle; margin-left:53px; margin-right:60px;z-index:100}
.thermostat div.absolute.bottom-0.text-center.w-full {color:rgba(128,128,0); outline:var(--Bk) dotted violet; z-index:-1}
/* End of Tile Builder Thermostat CSS */
```

7) Add Thermostat Tile



It will look like this:



8) Add Tile Builder Thermostat Tile



It will look like this:



9) Relocate the tiles using the tile positioning menu so they occupy the same space.

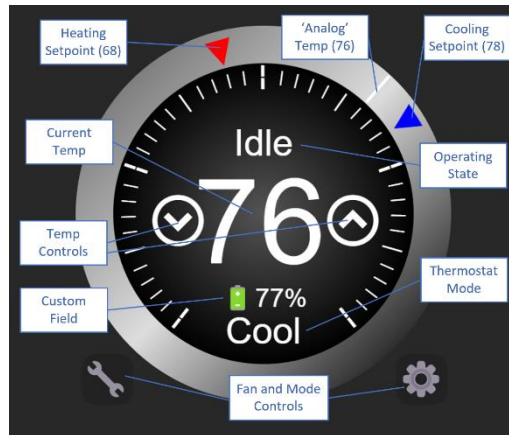
10) Finished, your result should look something like this.



Tile Builder Thermostat

Understanding the Display



The diagram below shows the various attributes of the display and their significance.






This diagram shows the display in Fahrenheit but Celsius is also supported.

Customizing Your Thermostat

As with all Tile Builder modules there is a significant level of customization available should you choose to do that. In this section I will just focus on the less obvious settings.

General Tab	Default Text Color: Affects the color of ALL text on the display including the gauge tick marks.
	Temperature Units: The default unit is Fahrenheit, but can be changed to Celsius. When using Fahrenheit, the vertical position marks 70 degrees and the gauge operates in the range 50 - 90 with marks at each of the 1, 5 and 10 degrees. When using Celsius, the vertical position marks 20 degrees and the gauge operates in the range 0 - 40 with marks at each of the 1, 5 and 10 degrees.
Display Tab	Thermostat Skin: This is the color\gradient of the outer circle. Options are: Black, Copper, Gold, Rose and Silver. Shown is Rose with Mark Ring setpoints. 
	Display Heating, Cooling Setpoints: The location and the display style of the Heating and Cooling Setpoints can be modified. Shown is Mark and Temp followed by Mark . 




Tile Builder Thermostat

	<p>Display Mode and Fan Controls: By default, these are visible but can be hidden if desired.</p> 
	<p>Display Additional Attribute: Allows you to display any additional attribute from your Thermostat in the gap below the temperature. For example, my thermostat has a battery backup.</p> <p> Display Additional Attribute: <input type="text" value="battery"/> Prepend Text: <input type="text" value=""/> Append Text: <input type="text" value=""/> Attribute Text Size: <input type="text" value="Small"/> </p> <p>By prepending and appending text, the result looks like this.</p>  <p>You can also just enter some static text by leaving the displayed attribute at None and entering some text into the dialog boxes.</p> <p> Display Additional Attribute: <input type="text" value="None"/> Prepend Text: <input type="text" value="Office"/> Append Text: <input type="text" value=""/> Attribute Text Size: <input type="text" value="Small"/> </p> 
Heating, Cooling, Idle, Other Modes	<p>Allows you to configure the appearance of the Thermostat background when the Thermostat changes modes.</p>

Tile Builder Thermostat

Testing Thermostat Modes

Rather than force your Thermostat mode to change you can preview the various mode settings by selecting a simulated mode.

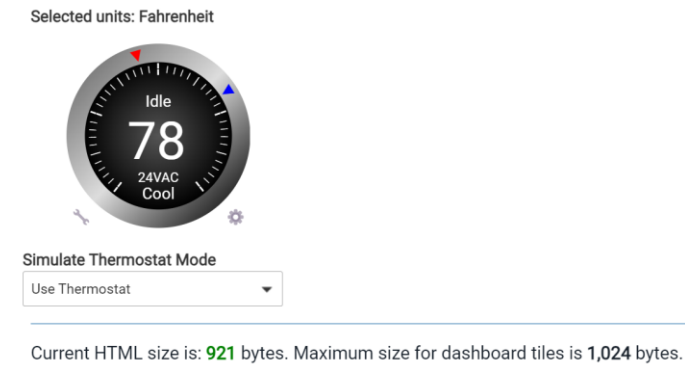
Simulate Thermostat Mode	Simulation Examples
<div>Use Thermostat ▼</div> <div>No selection</div> <div>Use Thermostat</div> <div>Simulate Cool - Cooling</div> <div>Simulate Heat - Heating</div> <div>Simulate Any - Idle</div> <div>Simulate Auto - Heating</div> <div>Simulate Off</div> <div>Simulate Emergency Heat - Heating</div> <div>Fahrenheit Calibration</div> <div>Celsius Calibration</div>	<div> Simulate Thermostat Mode Simulate Auto - Heating ▼</div> <div> Simulate Thermostat Mode Simulate Cool - Cooling ▼</div> <div> Simulate Thermostat Mode Simulate Off ▼</div>

When you are done be sure to set the simulation back to **Use Thermostat** so that you are getting live data.

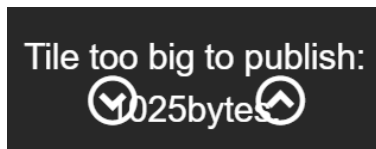
Tile Builder Thermostat

Managing Tile Size

Tile Builder Thermostat requires that the size of your configuration is less than 1,024 bytes and always display the current size of your configuration on screen. This ensures that your Thermostat tiles will work within the Hubitat App interface regardless of whether you are connecting locally or via the internet.



If your tile exceeds 1,024 bytes it will appear like this on the dashboard.



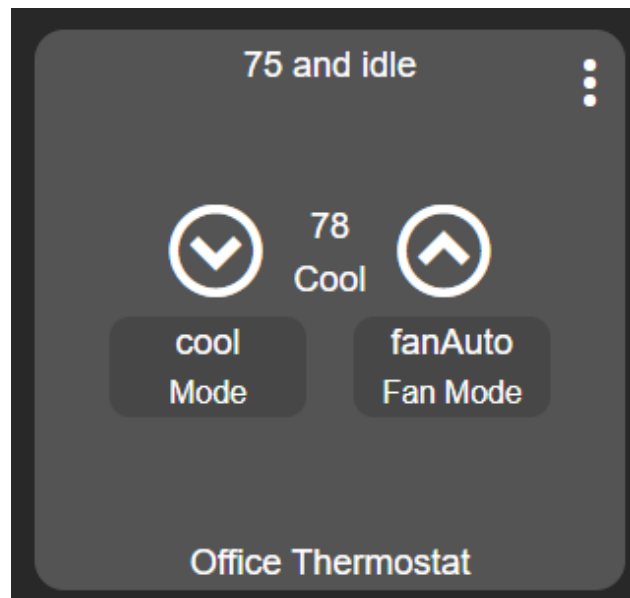
Tile Builder Thermostat

Troubleshooting

Slow Response Times

Issue: My Thermostat seems slow to respond to temp up\down commands.

With the existing thermostat tile the central part (78.0 Cool) shows the desired state. Pressing the down button 3 times in a row would result in the display going from 78 – 75. Each click of the down arrow sends a setCoolingSetpoint command to the Thermostat.



Depending on the design of your Thermostat one of two things may happen

- 1) The driver updates the coolingSetpoint attribute in the driver to the new value and then sends the command to the thermostat. This works very smoothly and quickly.
- 2) The driver sends a request to the thermostat for the new coolingSetpoint and waits for the thermostat to acknowledge the change request and **then** updates the attribute in the driver. This tends to be visually choppy and slow.

Bottom line the existing Thermostat tile shows the aspirational values as soon as a click is received. The Tile Builder only shows the actual value once the desired coolingSetpoint has actually been set.

Tile Builder Thermostat

Missing Controls

Issue: My thermostat control does not have buttons for controlling the Fan and/or thermostat mode.

Your driver is missing information that the Dashboard uses to determine how to display the tile.

To verify this, have a look at the attributes with the driver and make sure that the supportedThermostatFanModes and supportedThermostatModes information is present and is formatted as JSON with quotes and commas separating the values.

Current States

- coolingSetpoint : 75.0
- heatingSetpoint : 68.0
- hysteresis : 0.5
- supportedThermostatFanModes : ["auto","circulate","on"]
- supportedThermostatModes : ["auto","cool","emergency heat","heat","off"]
- temperature : 68.0
- thermostatFanMode : auto
- thermostatMode : off
- thermostatOperatingState : idle
- thermostatSetpoint : 68.0

If you have access to the driver code, you can execute these two statements to create these entries.

```
sendEvent(name: "supportedThermostatModes", value: ['"heat"', '"cool"', '"auto"', '"off"', '"emergency heat"'] )
sendEvent(name: "supportedThermostatFanModes", value: ['"auto"', '"circulate"', '"on"'] )
```

If this is a built-in driver you will need to report the issue to Hubitat support services.

Tweaking Publishing Performance

As with other Tile Builder modules Thermostat has an Event Timeout control. This control sets the minimum amount of time to wait before publishing the tile. In Thermostat the default is 250ms.

Tile Attribute to store the table? *	Name this Tile*	For Reference Only: Tiles in Use	Event Timeout (millis)
tile5	Office Thermostat - Tile 5	Click to set	0

If the eventTimeout is set to 0 (as shown) Tile Builder will publish a new Tile immediately on every change. However, if you were to click the down arrow 5 times in 1 second it would still publish the Tile 5 times. If you were to set the Event Timeout to 250ms it would wait until all 5 were received and then publish just once. What is best depends on your own situation and how responsive your thermostat is to temperature change requests.