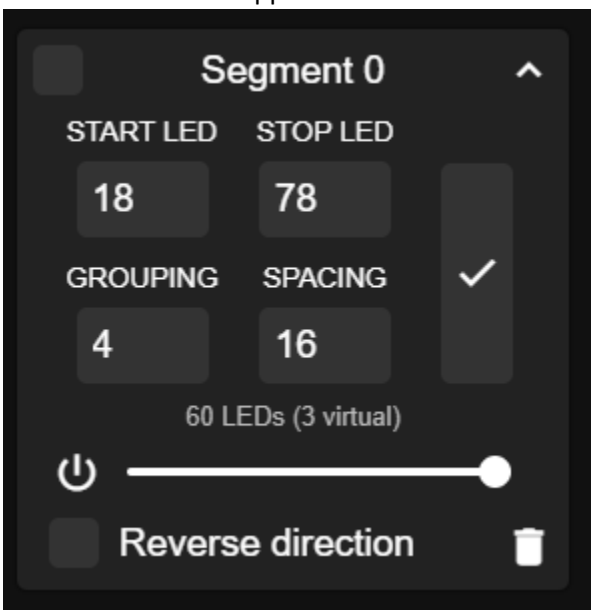
Creating Your Own Animations

Feel free to replace any of my animations with your own, or add your own to what is there. If you do come up with new animations my only ask is that you share them with me if you're willing. If I like yours better than my own I may add them to the official code release (with your permission).

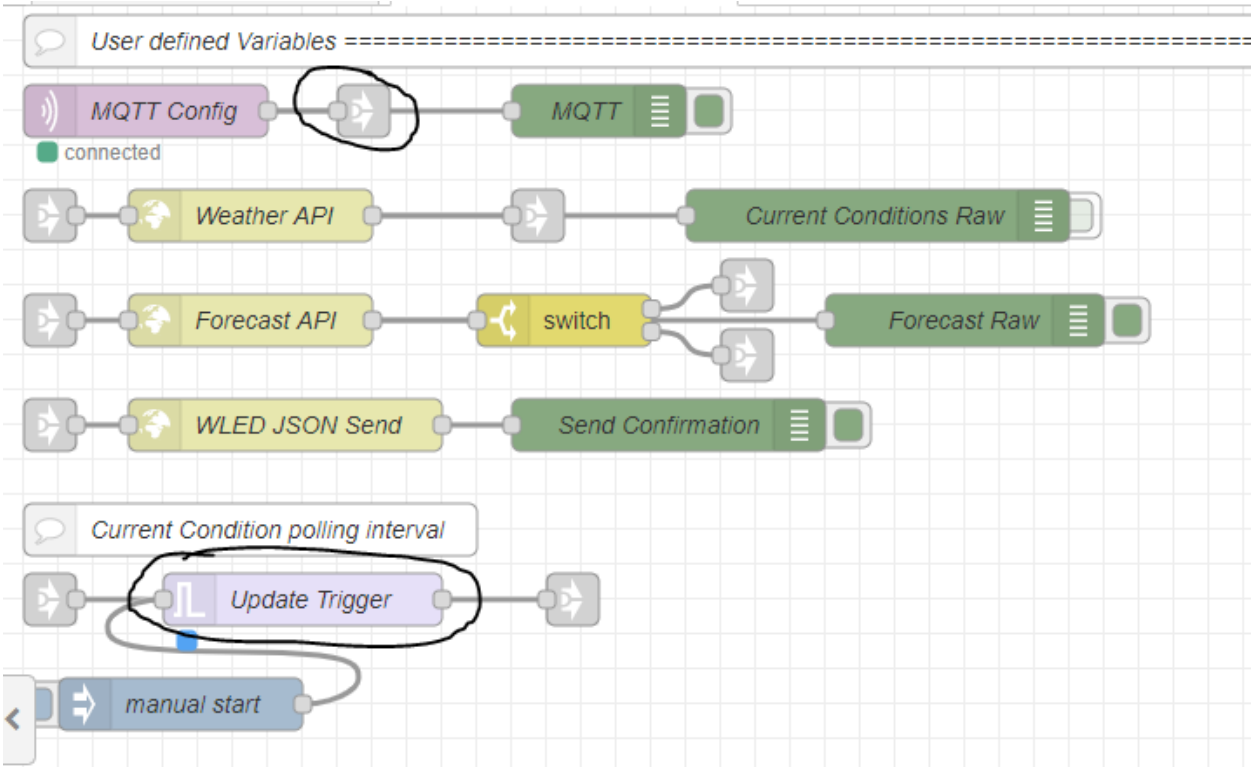
The animations are designed in the WLED controller page and then captured as their JSON data. There are a couple rules for animations that you must follow if you wish for them to work with the existing Node Red code.

0. Read the official WLED page on segments first - https://github.com/Aircoookie/WLED/wiki/Segments
1. Segment 0 must always be the band for temperature color. This is what the code currently assumes and if you change this it can cause problems. Segment zero is defined as follows with the solid animation applied:



2. you should never try to assign individual LEDs to more than one segment. Doing so causes the animations to glitch.
3. There are only 10 segments available, starting with segment 0 and ending with segment 9. That means that there are 9 segments available for you to create your animation since segment 0 is always the color band.
4. When documenting your animations, you will need to add definitions for any of the 9 segments you do not use that effectively delete them. This is because if your animation only uses 5 segments, but the one that was previously displayed had 8, it will apply your new 5 segments, but leave the extra 3 and cause glitches. More below on how to define the extra segments to delete them.
5. Lastly you should disable the auto update nodes of the node red code to prevent it from forcing the lamp to update as you are experimenting. To do this, disable the link out node attached to the MQTT Config Node in the Setup Section at the top, along with the Update Trigger Node in the same section. This should stop the Flow from responding to MQTT triggers, as well as stop

the periodic condition update trigger. Remember to turn these back on when you are ready for the lamp to resume normal use.



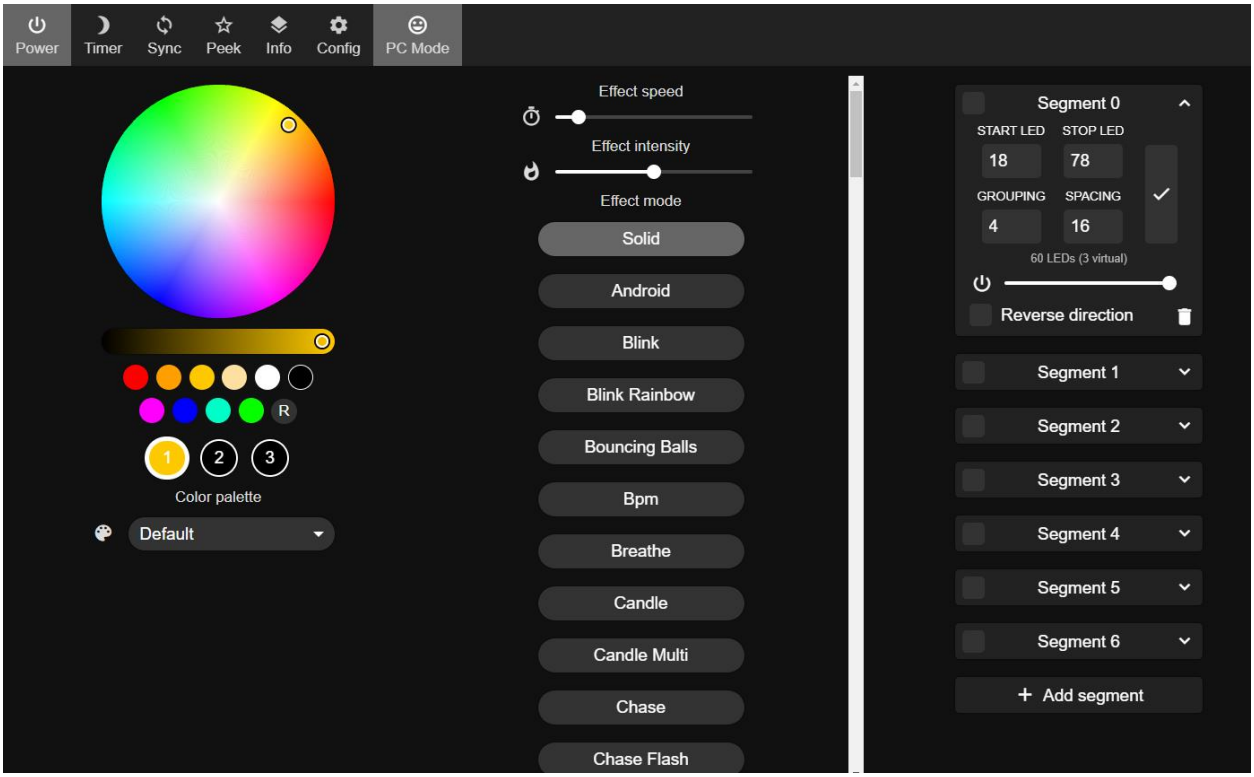
Creating Animations

Once you've read the WLED page on segments, animations are created by defining your segment of LEDs and then applying an effect to them. See the official page for descriptions of all the effects https://github.com/Aircoookie/WLED/wiki/List-of-effects-and-palettes

The Start LED is the first LED you want in the segment, and the stop LED is the number AFTER the last LED. Remember that WLED starts counting at 0, so the first LED on strand 1 is actually LED 0, and in this lamp the last led Is actually 79. The stop LED can be confusing because it defines the LED literally where the animation will no longer continue on to, not the last LED in the segment, which is why you need to define the number after the last LED you want lit. So if you want the first 4 LEDs on strand 1, you would start on LED 0 and end on LED 4, giving you LEDs 0, 1, 2, and 3 lit as that segment.

Grouping and spacing allow you to define groups of LEDs that skip along the line. You tell it how many LEDs to group together and then how many to skip before starting the next group. For example if you wanted the top 2 LEDs on each strand to be defined in a segment you would say Start LED 8, end LED 72, grouping 4, spacing 16. A very important note here is that many animations will NOT work when you have a spacing defined. They will glitch and kind of just flick between the two colors instead of the expected animation. Keep this in mind when laying out your segments.

Reverse direction allows you to run the animation in the other direction, which is helpful since every other strip in the lamp runs down instead of up.



Capturing the Animation JSON Data

When you have created your animation you need to capture the json data used to recreate it later on. This is because WLED cannot store more than one save state with multiple segments, so we need to save them in node red and apply them each time. To do this, load a new page in a web browser and then to go <your lamp IP address>/json. If your lamp is at 192.168.0.10, load 192.168.0.10/json. What you will see is the json string that sets the lamp exactly as you have it defined. The output will look like this:


```
{"state":{"on":true,"bri":128,"transition":7,"ps":-1,"pss":7,"pl":-1,"ccnf":{"min":1,"max":5,"time":12},"nl":
{"on":false,"dur":60,"fade":true,"tbri":0},"udpn":{"send":false,"recv":true},"lor":0,"mainseg":0,"seg":
[{"id":0,"start":18,"stop":78,"len":60,"grp":4,"spc":16,"on":true,"bri":255,"col":[[255,0,0],[0,0,0],
[0,0,0]],"fx":0,"sx":23,"ix":128,"pal":0,"sel":true,"rev":false},
{"id":1,"start":2,"stop":80,"len":78,"grp":6,"spc":4,"on":true,"bri":255,"col":[[0,106,255],[0,174,232],
[0,0,0]],"fx":2,"sx":23,"ix":0,"pal":0,"sel":false,"rev":false},
{"id":2,"start":28,"stop":32,"len":4,"grp":1,"spc":0,"on":true,"bri":255,"col":[[240,255,145],[0,0,0],
[0,0,0]],"fx":0,"sx":0,"ix":0,"pal":0,"sel":false,"rev":false},{"id":3,"start":8,"stop":12,"len":4,"grp":1,"spc":0,"on":true,"bri":255,"col":
[[0,106,255],[0,174,232],[0,0,0]],"fx":0,"sx":0,"ix":0,"pal":0,"sel":false,"rev":false},
{"id":4,"start":48,"stop":80,"len":32,"grp":4,"spc":16,"on":true,"bri":255,"col":[[0,106,255],[0,174,232],
[0,0,0]],"fx":0,"sx":0,"ix":0,"pal":0,"sel":false,"rev":false},{"id":5,"start":0,"stop":2,"len":2,"grp":1,"spc":0,"on":true,"bri":255,"col":
[[0,106,255],[0,174,232],[0,0,0]],"fx":96,"sx":0,"ix":23,"pal":0,"sel":false,"rev":true},
{"id":6,"start":78,"stop":80,"len":2,"grp":1,"spc":0,"on":true,"bri":255,"col":[[0,106,255],[0,174,232],
[0,0,0]],"fx":96,"sx":108,"ix":255,"pal":0,"sel":false,"rev":false}]},"info":{"ver":"0.10.0","vid":2005030,"leds":
{"count":80,"rgbw":false,"wv":false,"pin":
[2],"pwr":989,"maxpwr":3000,"maxseg":10,"seglock":false},"str":false,"name":"WLED","udpport":21324,"live":false,"lm":"","lip":"","fxcount":103
,"palcount":52,"wifi":
{"bssid":"EC:08:6B:AE:66:6F","rssi":-63,"signal":74,"channel":3},"arch":"esp8266","core":"2_6_3","lwip":1,"freeheap":15200,"uptime":116181,"op
t":119,"brand":"WLED","product":"FOSS","mac":"f4cfa2d453cd"},"effects":[
"Solid","Blink","Breathe","Wipe","Wipe Random","Random Colors","Sweep","Dynamic","Colorloop","Rainbow",
"Scan","Scan Dual","Fade","Theater","Theater Rainbow","Running","Saw","Twinkle","Dissolve","Dissolve Rnd",
"Sparkle","Sparkle Dark","Sparkle+","Strobe","Strobe Rainbow","Strobe Mega","Blink Rainbow","Android","Chase","Chase Random",
"Chase Rainbow","Chase Flash","Chase Flash Rnd","Rainbow Runner","Colorful","Traffic Light","Sweep Random","Running 2","Red & Blue","Stream",
"Scanner","Lighthouse","Fireworks","Rain","Merry Christmas","Fire Flicker","Gradient","Loading","Police","Police All",
"Two Dots","Two Areas","Circus","Halloween","Tri Chase","Tri Wipe","Tri Fade","Lightning","ICU","Multi Comet",
"Scanner Dual","Stream 2","Oscillate","Pride 2015","Juggle","Palette","Fire 2012","Colorwaves","Bpm","Fill Noise",
"Noise 1","Noise 2","Noise 3","Noise 4","Colortwinkles","Lake","Meteor","Meteor Smooth","Railway","Ripple",
"Twinklefox","Twinklecat","Halloween Eyes","Solid Pattern","Solid Pattern Tri","Spots","Spots Fade","Glitter","Candle","Fireworks Starburst",
"Fireworks 1D","Bouncing Balls","Sinelon","Sinelon Dual","Sinelon Rainbow","Popcorn","Drip","Plasma","Percent","Ripple Rainbow",
"Heartbeat","Pacifica","Candle Multi"
],"palettes":[
"Default","* Random Cycle","* Color 1","* Colors 1&2","* Color Gradient","* Colors Only","Party","Cloud","Lava","Ocean",
"Forest","Rainbow","Rainbow Bands","Sunset","Rivendell","Breeze","Red & Blue","Yellowout","Analogous","Splash",
"Pastel","Sunset 2","Beech","Vintage","Departure","Landscape","Beach","Sherbet","Hult","Hult 64",
"Drywet","Jul","Grintage","Rewhi","Tertiary","Fire","Icefire","Cyane","Light Pink","Autumn",
"Magenta","Magred","Yelmag","Yelblu","Orange & Teal","Tiamat","April Night","Orangery","C9","Sakura",
"Aurora","Atlantica"
]}
```


Copy this string into your favorite text editor. There is a bunch of data here that you do not need or want. First, you can trim all data from the “info” definition to the end.

There is also data at the beginning in the state section that you do not want to be sending each time. This info defines on/off state transition settings, and a number of other data variables you don’t need to set each time the animation updates. If you leave this in then your animation will override things like the LEDs being off if you have double pressed the button to turn the lamp off for a while.

You can remove everything up to the “mainseg” section, making sure to leave a single “{“ preceding it. You should now have a data block that looks something like:


```
{"mainseg":0,"seg":[{"id":0,"start":18,"stop":78,"len":60,"grp":4,"spc":16,"on":true,"bri":255,"col":[[255,0,0],[0,
0,0],[0,0,0]],"fx":0,"sx":23,"ix":128,"pal":0,"sel":true,"rev":false},{"id":1,"start":2,"stop":80,"len":78,"grp":6,
"spc":4,"on":true,"bri":255,"col":[[0,106,255],[0,174,232],[0,0,0]],"fx":2,"sx":23,"ix":0,"pal":0,"sel":false,"rev"
:false},{"id":2,"start":28,"stop":32,"len":4,"grp":1,"spc":0,"on":true,"bri":255,"col":[[240,255,145],[0,0,0],[0,0,
0]],"fx":0,"sx":0,"ix":0,"pal":0,"sel":false,"rev":false},{"id":3,"start":8,"stop":12,"len":4,"grp":1,"spc":0,"on":
true,"bri":255,"col":[[0,106,255],[0,174,232],[0,0,0]],"fx":0,"sx":0,"ix":0,"pal":0,"sel":false,"rev":false},{"id":
4,"start":48,"stop":80,"len":32,"grp":4,"spc":16,"on":true,"bri":255,"col":[[0,106,255],[0,174,232],[0,0,0]],"fx":0
,"sx":0,"ix":0,"pal":0,"sel":false,"rev":false},{"id":5,"start":0,"stop":2,"len":2,"grp":1,"spc":0,"on":true,"bri":
255,"col":[[0,106,255],[0,174,232],[0,0,0]],"fx":96,"sx":0,"ix":23,"pal":0,"sel":false,"rev":true},{"id":6,"start":
78,"stop":80,"len":2,"grp":1,"spc":0,"on":true,"bri":255,"col":[[0,106,255],[0,174,232],[0,0,0]],"fx":96,"sx":108,"
ix":255,"pal":0,"sel":false,"rev":false}]}
```

Lastly you will need to add sections for any unused segments as mentioned above to effectively delete them if they were previously defined. At the end of the data block, before the last “]” bracket, add the following for each unused segment

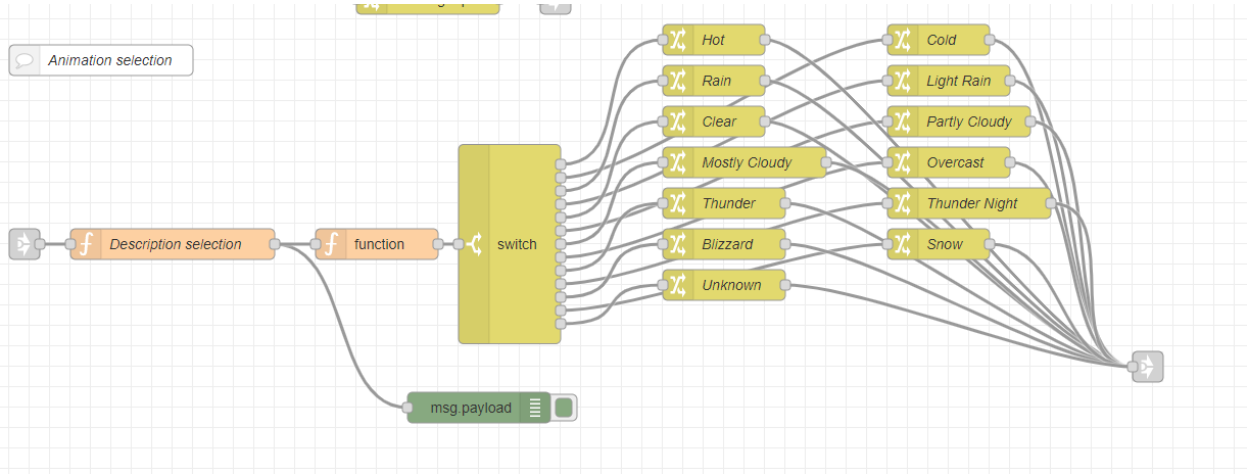
{"id":9,"start":0,"stop":0}

Be sure to change the id number to match the segment you are deleting, and add this line for EACH unused segment. If you only use segments 1-5 (leaving 0 for temperature), you need to add the above line for segments 6,7,8, and 9.

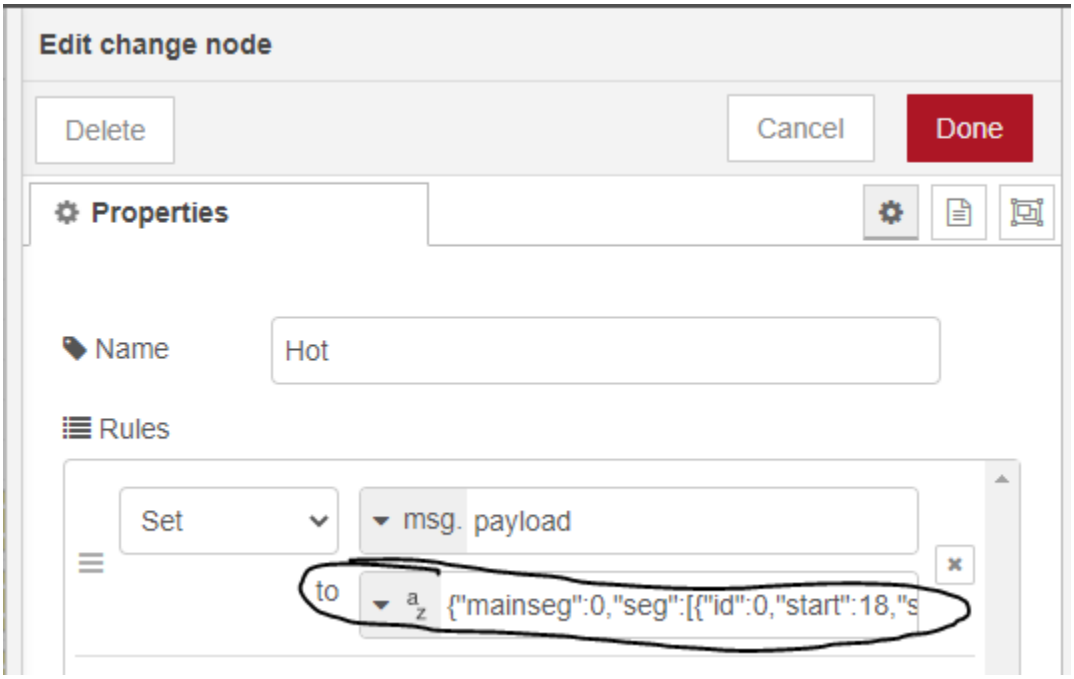
For reference, here is my animation json string for the clear/sunny animation:
{"mainseg":0,"seg":[{"id":0,"start":18,"stop":78,"len":60,"grp":4,"spc":16,"fx":0,"sx":23,"ix":128,"pal":0,"sel":false,"rev":false},{"id":1,"start":2,"stop":80,"len":78,"grp":6,"spc":4,"on":true,"bri":255,"col":[[0,106,255],[0,174,232],[0,0,0]],"fx":2,"sx":23,"ix":0,"pal":0,"sel":false,"rev":false},{"id":2,"start":28,"stop":32,"len":4,"grp":1,"spc":0,"on":true,"bri":255,"col":[[240,255,145],[0,0,0],[0,0,0]],"fx":0,"sx":0,"ix":0,"pal":0,"sel":false,"rev":false},{"id":3,"start":8,"stop":12,"len":4,"grp":1,"spc":0,"on":true,"bri":255,"col":[[0,106,255],[0,174,232],[0,0,0]],"fx":0,"sx":0,"ix":0,"pal":0,"sel":false,"rev":false},{"id":4,"start":48,"stop":80,"len":32,"grp":4,"spc":16,"on":true,"bri":255,"col":[[0,106,255],[0,174,232],[0,0,0]],"fx":0,"sx":0,"ix":0,"pal":0,"sel":false,"rev":false},{"id":5,"start":0,"stop":2,"len":2,"grp":1,"spc":0,"on":true,"bri":255,"col":[[0,106,255],[0,174,232],[0,0,0]],"fx":96,"sx":0,"ix":23,"pal":0,"sel":false,"rev":true},{"id":6,"start":78,"stop":80,"len":2,"grp":1,"spc":0,"on":true,"bri":255,"col":[[0,106,255],[0,174,232],[0,0,0]],"fx":96,"sx":108,"ix":255,"pal":0,"sel":false,"rev":false},{"id":7,"start":0,"stop":0},{"id":8,"start":0,"stop":0},{"id":9,"start":0,"stop":0}]}

Adding the new animation to the Node Red Flow

You will need to store this string in the node red flow and update the logic for when it gets called. If you are simply replacing one of the stock animations you will simply replace the string in the corresponding node. In the Animation Selection Section locate the animation node you wish to replace

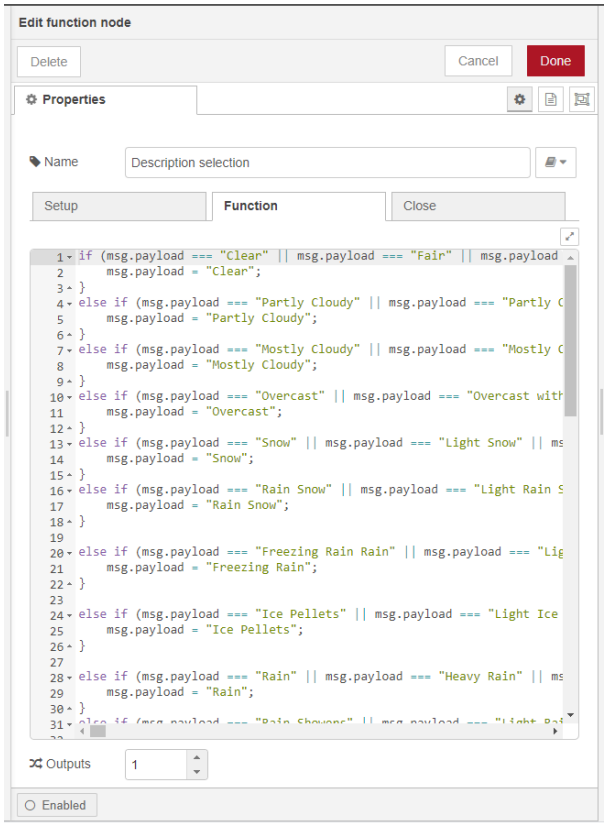


In the animation nodes, replace the string with your new json string, then click done, and redeploy the flow to update it.

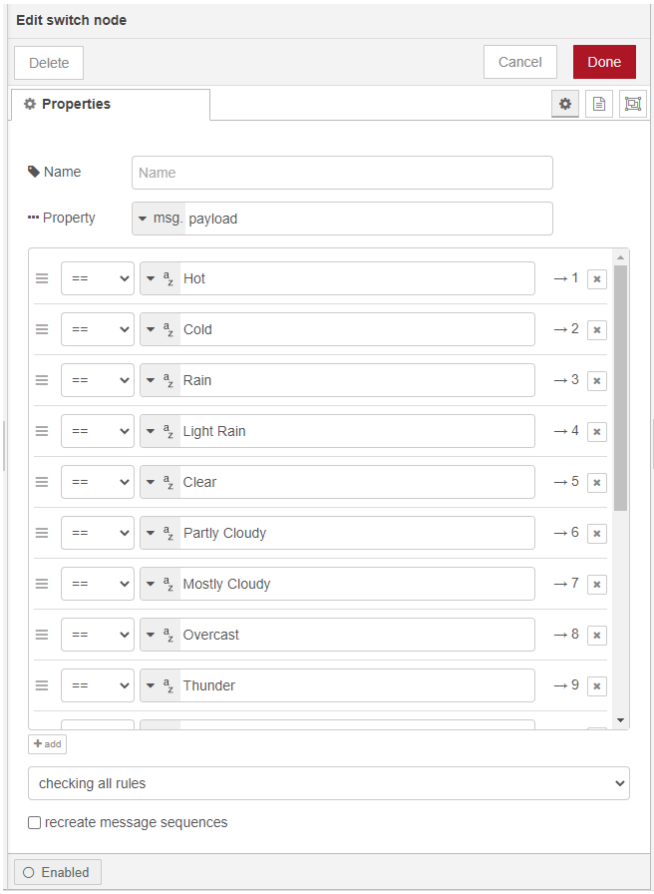


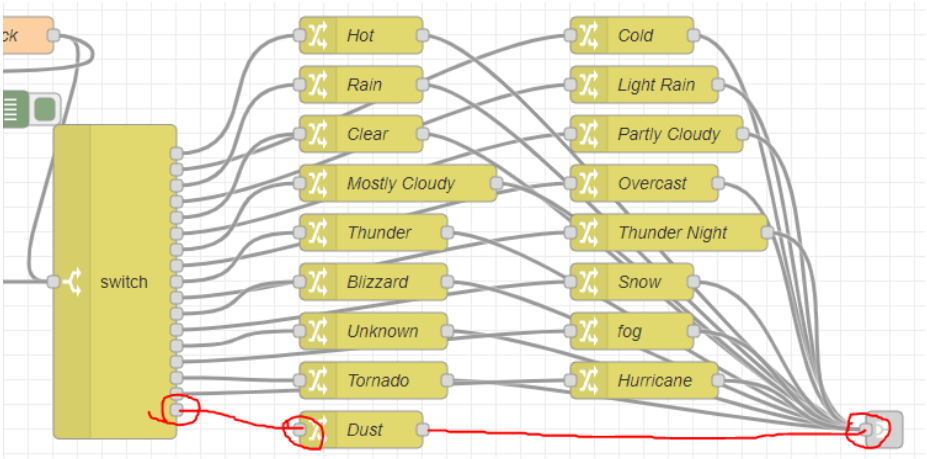
If you are adding a new animation you will have to update two other nodes, then add a new node with the json string. First, edit the Description Selection block. This contains a massive list of if/else if statements. The purpose here is to take the long list of simple descriptions that weather.gov uses, and group them into similar types that will use the same aminations. Note that currently there are many

more types in this list than there are finished animations. Search first to see if yours exists already. If not add another else if line that defines the description you want to define. Note that it MUST match exactly to what weather.gov publishes otherwise it will never be called.



Next, edit the switch node. You need to add the simple description that you defined or selected from the Description Selection node. Note that this must match EXACTLY, if you capitalize it in the Description selection node, capitalize it here. When you have added the new simple description, you will see a new linking dot appear on the switch node. Copy and paste one of the other animation change nodes, and link it to this new dot on the switch node. Then be sure to link the new Animation node to the link out node.





In the new animation change node you copied, replace the json string inside with your new one as described a couple steps ago. Deploy the changes you have made.