

Kevin Baugh

VBA Final Project: 10-Day Weather Report

### **Executive Summary**

Holland, a regional freight carrier with terminals throughout the Midwest, offers less-than-truckload (LTL) delivery in twelve states. In the freight industry, on-time delivery is key to ensuring successful operations and repeat business; however, in extremely cold weather conditions, trucks can experience serious performance issues if fuel gels up and plugs a truck's fuel filters – this can lead to breakdowns and more importantly, failed on-time delivery of customers' freight. The solution to this problem is to add fuel additives to a terminal's large fuel tanks before cold weather hits; this way, the fuel additives and fuel can properly mix before being dispensed into a truck's individual fuel tanks. This weather report helps terminal managers know when they need to add fuel additives to their terminal's fuel tanks by more easily anticipating extremely cold weather conditions. It also helps managers know if they have enough fuel additives on hand to handle the forecasted low temperatures for each location.

My report provides an automated solution for those at Holland to easily create a 10-day weather forecast for each of the company's locations. It's important for a terminal manager to know not only his or her terminal's weather conditions, but also the weather conditions of each truck's destination. This way, trucks can prepare for extremely cold weather conditions throughout the company's operational footprint. The report is currently formatted to provide weather data for each of the company's 53 locations on one worksheet. Users of the report can sort weather data by area, city, state, weather, chance of precipitation, high, and low. The source of the weather data is MSN Weather.

In addition to the report found on the worksheet labeled "Weather Report," there is a worksheet labeled "Personal Weather" that allows users to collect weather data for any U.S. city/state by simply entering information into cells A2 and B2 (highlighted). Although the purpose of this project was to solve Holland's weather forecasting needs (found on the Weather Report worksheet), anyone who needs to forecast the weather for specific locations can either change the city/state combinations on the Weather Report worksheet or use the Personal Weather worksheet to get weather data for new locations.

## Implementation

Sample view of the first two days for some of the company's locations:

	A	B	C	D	E	F	G	H	I	J	K
1	10-Day Weather Report			Saturday, April 9, 2016				Sunday, April 10, 2016			
2	Area	City	State	Weather	Chance of Precipitation	High	Low	Weather	Chance of Precipitation	High	Low
3	Area 1	Akron	OH	Mostly Cloudy	90%	33°	19°	Cloudy	80%	45°	32°
4	Area 1	Birch Run	MI	Partly Sunny	50%	31°	25°	Snow	100%	40°	33°
5	Area 1	Buffalo	NY	Cloudy	50%	29°	22°	Cloudy	100%	37°	30°
6	Area 1	Cincinnati	OH	Cloudy	100%	42°	28°	Cloudy	90%	56°	45°
7	Area 1	Cleveland	OH	Cloudy	90%	34°	23°	Cloudy	90%	48°	34°
8	Area 1	Columbus	OH	Partly Sunny	100%	34°	27°	Cloudy	80%	49°	38°
9	Area 1	Dayton	OH	Cloudy	100%	36°	27°	Cloudy	90%	51°	41°
10	Area 1	Detroit	MI	Cloudy	80%	32°	26°	Snow	100%	43°	31°
11	Area 1	Gaylord	MI	Mostly Cloudy	10%	25°	17°	Snow	100%	35°	27°
12	Area 1	Huntington	OH	Cloudy	100%	35°	26°	Cloudy	80%	49°	38°
13	Area 1	Jackson	MI	Mostly Cloudy	90%	30°	22°	Snow	100%	45°	30°
14	Area 1	Toledo	OH	Partly Sunny	100%	33°	23°	Snow	100%	47°	30°
15	Area 1	Youngstown	OH	Snow	100%	35°	17°	Cloudy	70%	42°	30°
16	Area 2	Atlanta	GA	Sunny	0%	58°	36°	Cloudy	10%	59°	44°
17	Area 2	Birmingham	AL	Sunny	0%	65°	42°	Sunny	0%	61°	44°
18	Area 2	Charlotte	NC	Sunny	0%	53°	34°	Mostly Cloudy	10%	58°	43°
19	Area 2	Chattanooga	TN	Sunny	0%	57°	36°	Mostly Cloudy	20%	63°	46°
20	Area 2	Columbia	SC	Sunny	0%	62°	39°	Partly Sunny	10%	61°	45°
21	Area 2	Decatur	AL	Mostly Sunny	0%	59°	37°	Sunny	0%	56°	39°
22	Area 2	Evansville	IN	Mostly Cloudy	0%	50°	33°	Sunny	0%	46°	36°
23	Area 2	Greensboro	NC	Partly Sunny	10%	52°	33°	Partly Sunny	0%	57°	47°
24	Area 2	Knoxville	TN	Sunny	40%	51°	32°	Mostly Sunny	30%	61°	48°
25	Area 2	Lexington	KY	Mostly Cloudy	80%	43°	29°	Cloudy	80%	57°	49°
26	Area 2	Louisville	KY	Mostly Cloudy	70%	45°	34°	Cloudy	90%	59°	52°
27	Area 2	Memphis	TN	Sunny	0%	64°	42°	Sunny	10%	61°	52°
28	Area 2	Nashville	TN	Partly Sunny	0%	58°	36°	Sunny	0%	53°	38°
29	Area 2	Raleigh	NC	Partly Sunny	10%	53°	34°	Partly Sunny	0%	57°	42°
30	Area 2	Spartanburg	SC	Sunny	0%	53°	34°	Partly Sunny	10%	59°	42°
31	Area 3	Chicago	IL	Rain Showers	80%	43°	28°	Sunny	80%	38°	33°
32	Area 3	Danville	IL	Light Rain	60%	42°	27°	Sunny	60%	40°	33°
33	Area 3	Fort Wayne	IN	Cloudy	100%	36°	26°	Cloudy	100%	50°	38°
34	Area 3	Grand Rapids	MI	Mostly Cloudy	100%	33°	26°	Snow	100%	45°	33°
35	Area 3	Indianapolis	IN	Cloudy	80%	41°	31°	Cloudy	100%	55°	47°

First, I inserted the agent class module (obtained from Gove Allen) into my workbook. After that, I entered each of Holland's 53 locations and accompanying area descriptions in columns A, B, and C. After getting the format of my report the way I wanted it, I then turned to <http://www.msn.com/en-us/weather> to view the page source to see how I could parse the HTML to get each location's weather data.

I experienced some difficulty using the `a.moveTo` command to capture the HTML elements I was trying to parse. After some trial and error, I was able to accurately parse through the HTML using the following code:

```

a.position = 1

a.moveTo ("The forecast for ")
a.moveTo ("title=""")
weather1 = a.getText("''''")
a.moveTo ("precipicn ")
a.moveTo (">")
a.moveTo (">")
precip1 = a.getText("</span>")
a.moveTo ("<p>")
high1 = a.getText("</p>")
a.moveTo (">")
low1 = a.getText("</p>")

```

I repeated this process for each day in the 10-day forecast (the example code above is only for the first day). After assigning values to each of my variables (i.e., weather, precipitation, high, low), I then inserted those values onto the worksheet by using the following code:

```

Cells(r, 4).Value = weather1
Cells(r, 5).Value = precip1
Cells(r, 6).Value = high1
Cells(r, 7).Value = low1

```

The process continues for each of the remaining nine days (the example code above is only for the first day). After each location's data was gathered and entered, I then moved on to the next city by using the following code (r is the active row):

```

r = r + 1

city = ActiveCell.Offset(i, 0).Value
state = ActiveCell.Offset(i, 1).Value

i = i + 1

If city = "" Then Exit Do

```

The process loops until there is no longer a city or state listed in the list of locations.

Note: the primary difference between the first and second worksheets is that the macro for the first worksheet (Weather Report) utilizes a loop while the second worksheet (Personal Weather) does not. On the second worksheet, the user must enter a new city/state combination in order to get new weather data. Once a new location has been added to the report and a user searches for a new location's weather data, the new data will then be added below any previous weather searches. Also, the macro will not run on the Personal Weather worksheet unless both a U.S. city and state have been entered into cells A2 and B2. If a user tries to run the code without a city or state, either of the following message box will pop up:

Microsoft Excel X

Please enter a valid U.S. city

OK

Microsoft Excel X

Please enter a valid U.S. state

OK

After finishing the weather reporting components of my project, I then created two ribbon tabs (Weather Report and Personal Weather) and added two buttons for each tab to run each report's two macros. Both tabs contain the same button icons but run different macros. To create each ribbon and button, I used Custom UI Editor for Microsoft Office. See screenshots below.



### Learning and Conceptual Difficulties

While working on this project, I did not encounter any major difficulties. It did take some patience to identify and properly code the correct elements within MSN Weather's HTML. It also required a thoughtful approach to ensure that the report's looping functionality was accurately identifying and pasting correct values for each row's city/state combination. One minor issue that I came across while working with the data was dealing with Fahrenheit and Celsius temperatures. MSN Weather provides both. In order to fix this, I had to set the "weadegreetype" equal to "F" when opening each page to ensure all temperatures were on the same scale. See screenshot below.

a.openpage "http://www.msn.com/en-gb/weather/today/" & city & "," & state & ",United-States/we-city?weadegreetype=F", True

Throughout this project, I learned of the power of using class modules. The agent class module used in this workbook makes scraping the web for data a relatively easy task once the code is successfully written.

**Assistance**

Gove Allen answered one question I had about why my Personal Weather macro was creating an “automation error” each time I tried to retrieve new weather data within a short period of time (it worked just fine if I waited 15-20 seconds before running again). He suggested that I move my agent declaration to the module level – this solved that error. Other than that, I did not receive any assistance for my project. As mentioned earlier, I did use Gove Allen’s agent class module.

**Other**

Sample screenshot of the Personal Weather worksheet (first two days only for Salt Lake City, UT):

	A	B	C	D	E	F	G	H	I	J
1	City	State	10-Day Personal Weather Report							
2	Salt Lake City	UT	Saturday, April 9, 2016				Sunday, April 10, 2016			
3	City	State	Weather	Chance of Precipitation	High	Low	Weather	Chance of Precipitation	High	Low
4	Salt Lake City	UT	Partly Sunny	30%	66°	52°	Partly Sunny	50%	67°	50°

Original source:

