

HCA Technical Note: Using Weather Underground with HCA

Before you can use Weather Underground (“WU”) as a weather provider you need to sign up as a developer. What WU is doing is making weather data available to people creating applications but as each of you are a “developer” of your own home design, you should have your own key.

Go here: <http://www.wunderground.com/weather/api/>

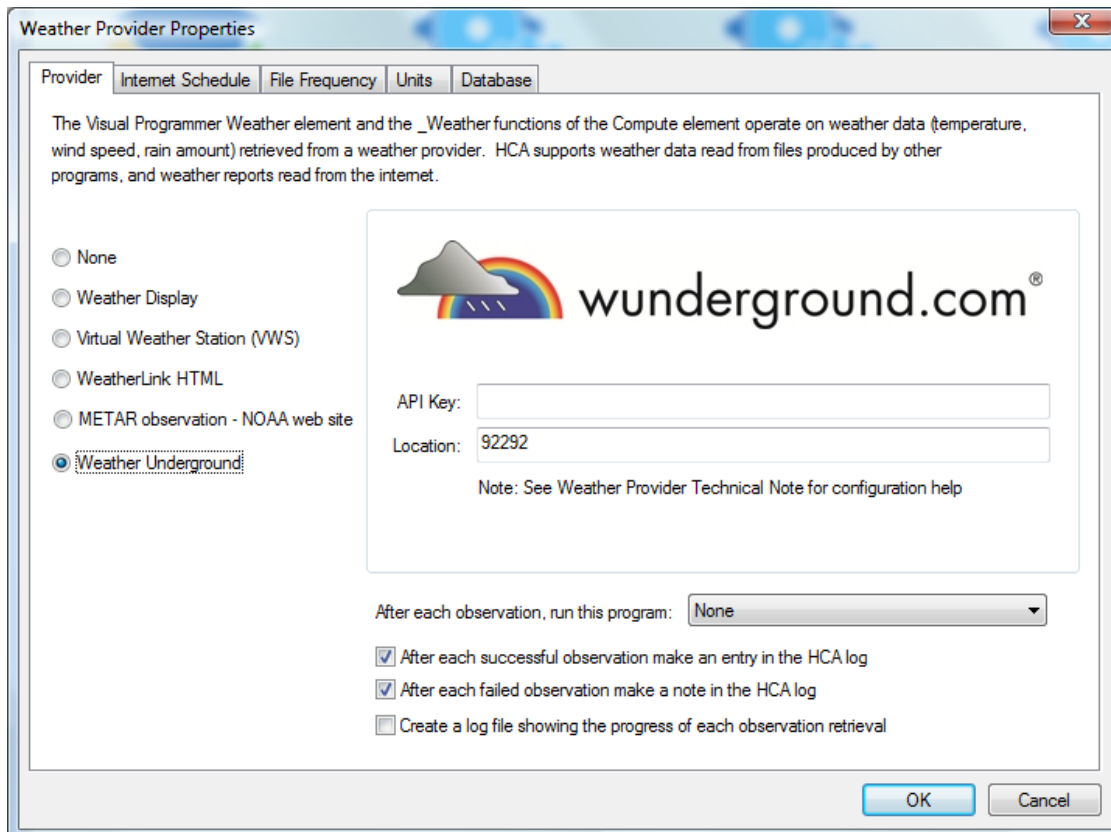
Complete the signup for the API Key. That is the key piece of info you need. You may also want to do a quick review of the documentation that WU provides. It may be helpful if you want to use the new features that go beyond the standard weather data extraction that HCA does.

Once you have your apikey, the next thing to do is to determine a location. Go here:

<http://www.wunderground.com/weather/api/d/docs?d=data/index>

Look at the section on “Query”. There are lots of different ways to specify your location. The easiest is to use your zip-code. One you have a key and a location you are ready to configure HCA.

In the HCA ribbon “Interfaces” category, “Weather” panel, press the “Provider Setup” button. On the “Provider” tab choose “Weather Underground” and enter your api key and location. No verification is done on either.



The screenshot shows the "Weather Provider Properties" dialog box with the "Provider" tab selected. The dialog contains the following elements:

- Provider:** Internet Schedule | File Frequency | Units | Database
- Description:** The Visual Programmer Weather element and the _Weather functions of the Compute element operate on weather data (temperature, wind speed, rain amount) retrieved from a weather provider. HCA supports weather data read from files produced by other programs, and weather reports read from the internet.
- Provider Selection:**
 - None
 - Weather Display
 - Virtual Weather Station (WVS)
 - WeatherLink HTML
 - METAR observation - NOAA web site
 - Weather Underground
- Weather Underground Configuration:**
 - API Key:
 - Location:
 - Note: See Weather Provider Technical Note for configuration help
- Observation Actions:**
 - After each observation, run this program:
 - After each successful observation make an entry in the HCA log
 - After each failed observation make a note in the HCA log
 - Create a log file showing the progress of each observation retrieval
- Buttons:** OK, Cancel

Some of the others tabs have changed from previous HCA versions. The “Internet Schedule” sets the schedule to be used to retrieve observations from WU and from METAR. The “File Frequency” tab is used for file based providers and has no effect on WU.

When using WU, on the units tab make sure that the “source” units are configured as:

- Temperature in “F”
- Rain in “Inches”
- Barometer in “Inches”
- Wind in “MPH”

When the schedule tells HCA to take an observation – retrieve data from WU - the XML is retrieved and saved in the “Temp” folder in the HCA documents area. If you choose to create a log file – last checkbox on the “Provider” tab – that is located in the “Logs” folder and is called “WeatherLog.txt”

After each observation is taken HCA extracts the data for these variables:

- Outside temperature
- Relative humidity
- Barometer (inches)
- Barometer Trend
- Wind speed (mph)
- Wind Dir (degrees)
- Wind Direction (string)
- Wind Chill
- Daily rain (precipitation today inches)
- Rain rate (1 hour Precipitation)
- Dewpoint
- Wind Gust (mph)
- WeatherCondition (string)

All these various values are extracted and are available to the Weather VP element as usual. You can also use the `_Weather` function to operate on these values.

One additional function has been added to allow you to get deeper into all the data provided by WU.

The function is called “`_WUnderground`” and it takes a minimum of 4 arguments and a max of 10. The format is:

```
_Wunderground ("feature", "location", "path1", "path2", ... "pathn", "tag")
```

The “feature” is one of these on this list:

<http://www.wunderground.com/weather/api/d/docs?d=data/index>

The location is any acceptable location as described on that page in the “query” section. You can use “” for the feature and it defaults to “conditions”. You can use “” for location and it defaults to whatever you entered as the location in the weather provider dialog.

Using this you can get to **anything** that weather underground provides. Here is an example:

Enter this into a browser:

<http://api.wunderground.com/api/<your api key>/forecast/q/<your zipcode>.xml>

Substitute your API key and location into this URL. This is similar to what you will see:

```

▼<response>
  <version>0.1</version>
  ▼<termsOfService>
    http://www.wunderground.com/weather/api/d/terms.html
  </termsOfService>
  ▼<features>
    <feature>forecast</feature>
  </features>
  ▼<forecast>
    ▼<txt_forecast>
      <date>7:00 AM PST</date>
      ▼<forecastdays>
        ▼<forecastday>
          <period>0</period>
          <icon>rain</icon>
          <icon_url>http://icons-ak.wxug.com/i/c/k/rain.gif</icon_url>
          <title>Wednesday</title>
          ▼<fcttext>
            ▼<![CDATA[
              Overcast with rain. High of 45F. Breezy. Winds from the South at 15 to 20 mph. Chance of rain 90%.
            ]]>
          </fcttext>
          ▼<fcttext_metric>
            ▼<![CDATA[
              Overcast with rain. High of 7C. Windy. Winds from the South at 25 to 30 km/h. Chance of rain 90%.
            ]]>
          </fcttext_metric>
          <pop>90</pop>
        </forecastday>
        ▶<forecastday>...</forecastday>
      </forecastdays>
    </txt_forecast>
  </forecast>

```

For this example, what I want to extract is the forecast string for today. The path to that string is this:

Response – forecast – txt_forecast – forecastdays – forecastday – fcttext

The `_wunderground` function that extracts this data is:

```
Forecast = _wunderground ("forecast", "", "forecast", "txt_forecast", "forecastdays",
"forecastday", "fcttext")
```

The top level of the path - “response” - is always used so you leave it off the path you create in the arguments to `_WUnderground`. Starting with the 3rd argument the path to the data wanted is given. The last argument is the tag that has data associated with it.

In the above example, the first argument tells it to retrieve forecast.xml. The second argument, given as an empty string, tells it to take the location from the weather provider setup. The next 4 are the path to the key we want. The last argument is the name key of the key with the data: in this case “fcttext”.

Given the above XML, this function evaluates to this string:

“Overcast with rain. High of 45F. Breezy. Wind from the South at 15 to 20 mph. Chance of rain 90%.”

Instead of the forecast, if you wanted to get the “icon” the path would be:

Response – forecast – txt_forecast – forecastdays – forecastday – icon

As a final example, to get the moon percent:

```
Moon = _wunderground ("astronomy", "", "moon_phase", "percentIlluminated")
```

There is one complication with some keys: In this example below there are multiple “forecastDay” branches:

```
▼<forecast>
  ▼<txt_forecast>
    <date>7:00 AM PST</date>
    ▼<forecastdays>
      ▶<forecastday>...</forecastday>
      ▶<forecastday>...</forecastday>
      ▶<forecastday>...</forecastday>
      ▶<forecastday>...</forecastday>
      ▶<forecastday>...</forecastday>
      ▶<forecastday>...</forecastday>
      ▶<forecastday>...</forecastday>
      ▶<forecastday>...</forecastday>
    </forecastdays>
  </txt_forecast>
```

In this case, include a qualifier in the name of the tag:

```
Forecast = _wunderground ("forecast", "", "forecast", "txt_forecast", "forecastdays",
"forecastday:3", "fcttext")
```

And it retrieves that specific forecast day. To get the first forecast day use “forecastday:1”. To find the second use “forecastday:2”, etc. If you leave off the qualifier it defaults to 1.

A good suggestion is to try out the various retrievals in your browser first then look at the data and figure out the path you need to get at that data in HCA. Pay close attention to any paths that will need qualifiers.

Two more important points:

To prevent multiple queries to Weather Underground each time a `_wunderground` function is evaluated, the file is retrieved, parsed, and saved internally along with the date/time that it was retrieved and parsed. Subsequent `_wunderground` functions don’t retrieve the data again from the internet if the already parsed XML is “recent”. There is a registry setting – currently no UI for it – that says how many

minutes the data stays “recent”. For now that is set to 15 minutes. This way you can extract lots of data and do it quickly since only one internet action is needed.

And finally, the interactions with the Weather Underground site are done using HTTP so there should be no firewall issues.