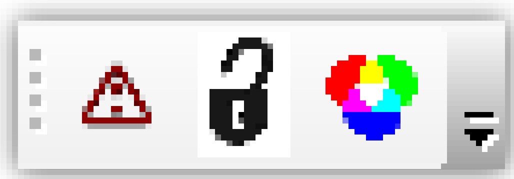
ArcGIS for Desktop Add-In Toolset for Retrieving and Displaying Natural Hazard Data



David Lindsey

GISC 6387 - GIS Workshop Project

2017 August 10th

Introduction

When it comes to working with Geographic Information System (GIS) data, automating redundant processes within a workflow can save a researcher large amounts of time and effort. If redundancy can be avoided, it should be avoided. As a GIS programmer, I thoroughly enjoy creating tools that enhance a workflow. With that said, this GIS Workshop project encompasses an extensive workflow that has been automated for interactive convenience.

I have opted to create an ArcGIS for Desktop Add-In toolset utilizing a .NET programming language (Visual Basic .NET, specifically) and ArcObjects. This toolset has been designed to benefit researchers and users who work with several natural hazard types. For the purposes of this project, the criteria for natural hazard data has been limited to earthquakes, hail, hurricanes, and tornadoes.

Both the National Oceanic and Atmospheric Administration (NOAA) and United States Geological Survey (USGS) maintain active data feeds on their websites for these hazard types. The data are provided and accessible via URL as downloadable CSV files and ESRI Shapefiles. Typically, a user is required to locate these websites, manually access the data URLs, download the raw CSV or Shapefile data, review the raw data for critical formatting errors, convert CSV files to Shapefile, assign a coordinate system, import the Shapefiles into ArcMap, and, finally, symbolize the data; all before any analysis or research can commence. Without automation, this process would cost the user a considerable amount of time and lead to potential human errors along the way.

Although this project's concept of automation is not revolutionary, its intent is to allow the user to quickly and efficiently access the desired natural hazard data with a few clicks of a mouse. Utilizing this toolset will allow users and researchers an improved workflow that includes improved time management.

Literature Review

When I set out to accomplish this project, I wanted to build an ArcGIS Add-In toolset that allowed for the most convenience to the user with regard to the natural hazard options, as well as ease of data conversion/loading within ArcMap. With that in mind, I searched through numerous sources in an attempt to find tools that were constructed similarly to this project idea. Although this project's Add-In toolset purpose sounded simplistic, I was unable to locate literature with an identical premise. This discovery fueled my interest in building this particular Add-In toolset. If consistently working with various types of natural hazard data, I feel that this Add-In toolset will be beneficial to a variety of ArcGIS for Desktop users.

While reviewing different literary sources during my research, I came across an article about a similar earthquake-related ArcGIS Add-In tool, entitled "SeismoGIS: A Tool for the Visualization of Earthquake Data" (Williams, Weskamm, Baaser, Hinzen, & Bareth, 2008). Within the paper, the authors described an ArcGIS Add-In tool that was programmed with .NET and ArcObjects. Its purpose was to allow European scientists the ability to quickly load earthquake data and provide analysis of the data by programmatically extracting all required, routine inputs from a raw ASCII file. For these scientists' purposes, this was a very effective tool. However, the tool was only useful for those users who had access to a proprietary software called SEISAN. SEISAN is, summed up, a software package that processes seismic activity from around the world (Ottemaller, 2014). Without the SEISAN software installed, the Add-In tool is useless.

Having said that, as robust as this tool appears to be for the scientific community, it does not appear to allow the user as much flexibility as it could. This tool does not pull the raw data from the internet, nor does it appear to allow the user to select a specific timeframe for historical data. The toolset that I am creating for this project will not require any additional requirements, aside from ArcGIS for Desktop (10.x) and an internet connection. Users of my toolset will be able to select specific historical timeframes for data loading, depending on the natural hazard selected (e.g. earthquakes from previous 24 hours versus previous 30 days, tornadoes from 2009 versus 2015, etc.). The Add-In Toolset being created for this project will be designed for ease-of-use for whoever should need to utilize it. Because the toolset itself is not complicated for the user, nor does it require any complex pre-setup to function, I feel that this will be an easier, straightforward option to be considered for quick analysis of various natural hazard data.

Study Area and Data Sources

As mentioned previously, the hazard data utilized for this project originated from NOAA and USGS. NOAA's website[1] maintains data access for hail, hurricanes, and tornadoes (among others), while the USGS website[2] maintains access for earthquake data. The hail and tornado data obtained from NOAA are specific to the United States geographic footprint. The hurricane data, as well as the earthquake data, are maintained on a global scale.

Concerning the data format, the earthquake, hail, and tornado data are provided as tabular CSV files. These files include numerous columns that specify such things as XY values, magnitude values, intensity, date of occurrence, etc. The hurricane data is accessible as CSV file

[1] http://www.spc.noaa.gov/wcm/#data
[2] https://earthquake.usgs.gov/earthquakes/feed/v1.0/csv.php

or Shapefile. Similarly, these Shapefiles include attribute data referencing wind speed values, date of occurrences, hurricane names (if applicable), among other values. For the purposes of this project, the hurricane's Shapefile data URLs[3] will be accessed by the toolset.

For the hail and tornado data, users (as of this report's date) have the ability to access information dating back to the 1950s. For hurricane data, users can access Shapefiles dating back to 1842. For earthquake data, the USGS maintains "live feeds" ranging from **Past Hour**, **Past Day**, **Past 7 Days**, and **Past 30 Days**. Within those timeframes, users can specify earthquake magnitude values ranging from **All**, **1.0+**, **2.5+**, and **4.5+**. All of these variables would be utilized as dynamic parameters within the toolset's capabilities.

Methodology

In order to consider and build the desired Add-In toolset, certain prerequisites needed to be met. Access to the URL source files via USGS and NOAA websites needed to be verified. Once all URL data sources were verified and CSV/Shapefile data were successfully downloading, the Add-In toolset could be constructed.

In order to build the Add-In toolset, several software components were required. For this project's purposes, I operated on Microsoft Windows 10 using Microsoft Visual Studio 2015. Visual Basic .NET (VB.NET) and ArcObjects were utilized and deployed within ArcGIS for Desktop 10.4.1 (Advanced licensing level). For research and implementation use, numerous physical and online reading resources were helpful and, ultimately, necessary to accomplish all desired objectives[4].

[3] ftp://eclipse.ncdc.noaa.gov/pub/ibtracs/v03r09/wmo/csv/year
[4] Please see Reference section for specific sources utilized.

The project parameters that were constructed to solve the initial problem were prioritized, structured, and implemented as follows:

- One ArcGIS for Desktop premier **Add-In Toolbar**, consisting of three individual **Add-In Buttons** (see **Appendix – Figure 1**).
    1. **Add-In Button** for retrieving user-specified hazard data from the web, converting the data into Shapefile, importing into ArcMap, and symbolizing it. This Hazard Data Button is coded as "btn_HazardDataMenu" with a Toolbar graphic of a hazard sign (leftmost icon on the Toolbar).
    2. **Add-In Button** for removing Shapefile locks from an ArcMap session. This Lock Release Button is coded as "btn_LockRelease" with a Toolbar graphic of an unlocked lock (middle icon on the Toolbar).
    3. **Add-In Button** for symbolizing hazard Shapefiles that were manually added to ArcMap by user. This Symbology Button is coded as "btn_Symbology" with a Toolbar graphic of a color wheel (rightmost icon on the Toolbar).

Operation-specifics for the Hazard Data Button:

Once this **Add-In Button** is clicked, a Windows Form dialog box will appear and the user will be prompted to select the desired natural hazard (see **Appendix – Figure 2**). Once the user selects the hazard type, another dialog box will be open with unique parameters specific to the chosen hazard (see **Appendix – Figures 3, 4, 5, and 6**). Inside the code, these four dialog boxes are stored/referenced as "frm_earthquakeOptions", "frm_hailOptions", "frm_hurricaneOptions", and "frm_tornadoOptions".

Within these parameter windows, the code relies on Combo Box drop-down list options for the user. These drop-down lists include numeric *timespans* (e.g. year), as well as *magnitudes* (for earthquakes only). The values available correlate to the data available on the NOAA and USGS websites. As of the date of this publication, here are the breakdowns of each hazard's data availability:

For Earthquakes:

- USGS Timespan choices are:
    - Past Hour, Past Day, Past 7 Days, Past 30 Days
- USGS Magnitude choices are:
    - All, 1.0+, 2.5+, 4.5+

For Hail:

- NOAA Timespan choices are:
    - All | 1955-2016, 2016, 2015, 2014, 2013, 2012, 2011, 2010, 2009, 2008, 2005-2007, 2000-2004, 90-99, 80-89, 70-79, 60-69, 50-59

For Hurricanes:

- NOAA Timespan choices are:
    - All | 1842-2016, as well as every individual year from 1842 to 2016.

For Tornadoes:

- NOAA Timespan choices are:
    - All | 1950-2016, 2016, 2015, 2014, 2013, 2012, 2011, 2010, 2009, 2008, 2005-2007, 2000-2004, 90-99, 80-89, 70-79, 60-69, 50-59

Each hazard type has a partially unique, **static** web URL where all data is accessed from. Certain parameters in that URL string can be changed, via the timeframe and/or magnitude drop-down lists. The timespan/magnitude choices from above are assigned to **dynamic** variables within the code. Each variable is then concatenated within the static URL for whichever website possesses the CSV/Shapefile data. To ensure stability, each Combo Box drop-down list is read-only. The user is not allowed to manually enter timespan or magnitude values.

Example URL for Earthquakes (Earthquakes with **Magnitude 2.5+** during **Past Hour**):

https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/2.5_hour.csv

Example URL for Hail (Hail occurrences for **2016**):

http://www.spc.noaa.gov/wcm/data/2016_hail.csv[5]

Example URL for Hurricanes (Hurricane occurrences for **2016**):

ftp://eclipse.ncdc.noaa.gov/pub/ibtracs/v03r09/all/shp/year/Year.2016.ibtracs_all_lines.v03r09.zip[6]

Example URL for Tornadoes (Tornado occurrences for **2016**):

http://www.spc.noaa.gov/wcm/data/2016_torn.csv[7]

To allow the user the ability to review the URL string being created, and to troubleshoot if necessary, a read-only textbox has been created within each hazard's parameters dialog box. This textbox will automatically populate the full URL once all Combo Box drop-down list parameters have been selected *(*see **Appendix - Figure 7***)*. The next parameter for the user to

[5] If wanting to download all hail occurrences from **1955-2016**, the CSV is provided as a zipped folder. As a result, the URL string changes slightly to: **http://www.spc.noaa.gov/wcm/data/1955-2016_hail.csv.zip**
[6] If wanting to download all hurricane occurrences from **1842-2016**, the URL string changes slightly to: **ftp://eclipse.ncdc.noaa.gov/pub/ibtracs/v03r09/all/shp/Allstorms.ibtracs_all_lines.v03r09.zip**
[7] If wanting to download all tornado occurrences from **1950-2016**, the CSV is provided as a zipped folder. As a result, the URL string changes slightly to: **http://www.spc.noaa.gov/wcm/data/1955-2016_torn.csv.zip**

select involves a destination output folder location where the code will save the downloaded file. This consists of a Browse button that launches a dialog box to select an output folder location. Once the folder location has been chosen, the file path will automatically display within another read-only textbox (see **Appendix – Figure 8**). The user also has the ability to choose whether or not to automatically import the downloaded data into ArcMap and symbolize it. This option consists of a checkbox labelled "Add to ArcMap" (see **Appendix – Figure 9**). This could be useful for those individuals who may want to download the data/create the shapefile, but not need to work with it immediately.

Once all parameters have been selected, the user clicks the OK button. If any critical parameters are not chosen (timespan, magnitude, or output folder), the code will alert the user of this problem with a message to select all parameters. The code will not continue if this is not done. If all parameters are set when the user clicks OK, the code executes the download from the created web URL string to the output folder. So how does this process work, exactly?

Once the user clicks OK, a text message will appear on the dialog window to inform the user that the code is "…Processing…Please Wait…" (see **Appendix – Figure 10**). Next, once the code has established a connection with the URL, the data beings the download process. The status of this download will be available via a progress bar that shows the file name being downloaded, and the file location it is being downloaded to. Due to some naming convention issues related to the source data name structure, alterations are needed while downloading the data. Much of the original source data is vague and potentially confusing to work with if numerous data files are downloaded. For example, the CSV file downloaded for **All** earthquakes during the **Past 7 Days** is provided with the naming convention of "all_week.csv". This naming convention needs enhancement. Additionally, some data files that were downloaded included hyphens. For example, all hail data spanning 1955-2016 is provided with the naming convention of “1955-2016_hail.csv”. The ArcObjects geoprocessing tools are unable to create a shapefile with hyphens included in the naming convention. As a result, the hyphens needed to be altered. To accommodate these issues, the code alters the naming convention of the downloaded file as it is being downloaded. If the data is downloaded from NOAA, “noaa_” is added to the beginning of the downloaded file’s naming convention. If the data is downloaded from USGS, “usgs_” is added to the beginning of the downloaded file’s naming convention. Additionally, any hyphens are changed to “_to_”. Underscores do not cause problems with the geoprocessing functionality within the code. Lastly, to better assist the user with identifying when data was downloaded, the current date (YYYYmmdd) is added to the naming convention as well. These modifications are viewable within the progress bar as the file is being downloaded (see **Appendix – Figure 11**).

After the download finishes, a message box will appear that states that the download has completed, as well as the name of the folder location where the file is being saved (see **Appendix – Figure 12**). The code will then create a new folder within the user-defined output folder location and deposit the downloaded file within it. The naming convention will match the downloaded file (minus the file extension). The purpose for this is to keep all affiliated files grouped together. During the code’s entire process, numerous files get created (Shapefiles alone produce several of these files). These folders assist in minimizing clutter and preventing some output files from being deleted by accident. Once the downloaded file has been saved/extracted within the newly created folder, any zipped folders that were originally downloaded get deleted by the code (again, this is to minimize clutter).

Next, the code goes into a data review process. If the downloaded file is a CSV, the code reviews that file for discrepancies that could cause the application to fail. After extensive tests and investigations, data inconsistencies for various timespans were evident. Some CSV files downloaded may be void of any data (e.g. earthquakes within the past hour with magnitude 4.5+ may not occur every hour, so these CSV files would not have earthquake entries). Some files had headers, while other CSV files did not. Some CSV files possessed 0,0 Lat/Long values. If the headers were missing, then attribute field names could not be assigned for the shapefile conversion or symbolization process. If zeroes existed in the XY fields, then the Shapefile output would contain erroneous data points. To combat these problems, the code was programmed to review the downloaded CSV for emptiness as well as missing headers. If the file possessed zero data rows (excluding header rows), the user would be notified via a pop-up message stating this occurrence. If the headers were missing, it would assign the headers for the user. If the XY columns contained zeroes, the code would remove those rows from the CSV entirely. To accomplish this in VB.NET, a new CSV needed to be created. As the code reviews the original CSV file (via StreamReader), it performs any manipulations and writes the modifications to a new CSV file (via StreamWriter) (see **Appendix – Figure 13**). This new CSV file contains the naming convention of the original CSV file, with "_checked" added to the end of the file name. Once this CSV review process is completed, the code will notify the user via a pop-up message box (see **Appendix – Figure 14**). With the CSV corrected, the code can proceed to convert the checked CSV file into a Shapefile.

The Shapefile conversion process only applies to the Earthquake, Hail, and Tornado data, as those hazards are provided as CSV files. The Hurricane data is downloaded as a Shapefile, and therefore does not need to go through this conversion process. With that said, the code takes the checked CSV file and creates temporary XYEvents. For Hail and Tornado data, the X values were assigned with "slon" and the Y values were assigned with "slat" header information. For the Earthquake data, the X value was assigned with “longitude” and the Y value was assigned with “latitude” header information. All XYEvents were assigned with geographic coordinate system WGS84 for accuracy and uniformity. After the temporary XYEvents were created, the geoprocessing tool “FeatureClasstoShapefile” was utilized to export the data to Shapefile. All Shapefile naming conventions were designed to be identical to the checked CSV file. With the Shapefile export completed, the folder where all of this data is now stored should include: the original CSV file that was downloaded from the web URL, the checked CSV, and all files affiliated with the Shapefile export (see **Appendix – Figure 15**).

If the user selected the “Add to ArcMap” checkbox, the code would proceed to automatically import the Shapefile into ArcMap with proper symbology assigned. The code is assigned with the current classification/color structure below.

For Earthquakes (see **Appendix – Figure 16**):

- Marker Type: SimpleMarkerSymbol
- Marker Style: esriSMSCircle
- Attribute Field Classified: “mag” (earthquake magnitude)
- Class Breaks: 3
    - Earthquakes with Magnitudes “Less than 3.0” (Color: Green, Size: 10)
    - Earthquakes with Magnitudes “Between 3.0 and 5.9” (Color: Orange, Size: 15)
    - Earthquakes with Magnitudes “6.0 and Above” (Color: Red, Size: 20)

For Hail (see **Appendix – Figure 17**):

- Marker Type: SimpleMarkerSymbol
- Marker Style: esriSMSCircle
- Attribute Field Classified: “mag” (hail diameter in inches)

- Class Breaks: 4
    - Hail with Diameters “Less than 1.0” (Color: Green, Size: 4)
    - Hail with Diameters “Between 1.0 and 1.99” (Color: Dark Yellow, Size: 6)
    - Hail with Diameters “2.0 and 2.99” (Color: Orange, Size: 8)
    - Hail with Diameters “3.0 and Above” (Color: Red, Size: 10)

For Hurricanes (see **Appendix – Figure 18**):

- Line Type: SimpleLineSymbol
- Line Style: esriSLSSolid
- Attribute Field Classified: “wmo_wind” (wind speeds in miles per hour)
- Class Breaks[8]: 8
    - Category with Wind Speeds “Unspecified” (Color: Black, Line Width: 1)
        - Unspecified includes erroneous wind speeds, whether missing or negative.
    - Category with Wind Speeds from 0-38.99 mph
        - “Tropical Depression” (Color: Light Green, Line Width: 1.5)
    - Category with Wind Speeds 39-73.99 mph
        - “Tropical Storm” (Color: Green, Line Width: 1.5)
    - Category with Wind Speeds 74-95.99 mph
        - “Category 1” (Color: Dark Yellow, Line Width: 1.5)
    - Category with Wind Speeds 96-110.99 mph
        - “Category 2” (Color: Orange, Line Width: 1.5)
    - Category with Wind Speeds 111-129.99 mph
        - “Category 3” (Color: Dark Orange, Line Width: 1.5)
    - Category with Wind Speeds 130-156.99 mph
        - “Category 4” (Color: Red, Line Width: 1.5)
    - Category with Wind Speeds 157+ mph

[8] Class breaks are based on official NOAA categories: http://www.aoml.noaa.gov/hrd/tcfaq/A5.html

        - "Category 5" (Color: Pink, Line Width: 1.5)

For Tornadoes (see **Appendix – Figure 19**):

- Marker Type: SimpleMarkerSymbol
- Marker Style: esriSMSCircle
- Attribute Field Classified: "mag" (tornado EF value)
- Class Breaks: 7
    - Tornado with Classification "Unspecified" (Color: Black, Size: 4)
        - Unspecified includes erroneous wind speeds, whether missing or negative.
    - Tornado with Classification "EF-0" (Color: Green, Size: 6)
    - Tornado with Classification "EF-1" (Color: Dark Yellow, Size: 8)
    - Tornado with Classification "EF-2" (Color: Orange, Size: 10)
    - Tornado with Classification "EF-3" (Color: Dark Orange, Size: 12)
    - Tornado with Classification "EF-4" (Color: Red, Size: 14)
    - Tornado with Classification "EF-5" (Color: Pink, Size: 16)

Once the code assigns all symbology to the imported Shapefiles, the Shapefile data will be easily recognizable within ArcMap (see **Appendix – Figures 20, 21, 22, and 23**). Upon completion of the symbology process, the parameters dialog box with the "…Processing…Please Wait…" message will automatically close. This dialog box closure is what informs the user that the process has completed.

As many ArcMap users are painfully aware, when a Shapefile gets imported into ArcMap, a new "lock" file is established on the Shapefile (see **Appendix – Figure 24**). Sometimes this lock will persist, even after the Shapefile has been removed from ArcMap.

Having said that, what happens if a user downloads/imports a Shapefile, then attempts to download/import the same exact data again? Luckily, the code is designed to account for these potential occurrences. When a user attempts to download data that is identical to data previously downloaded, the code will attempt to delete the original and overwrite it with the new data being downloaded. There are several steps that the code will take once it recognizes that duplicate data exists:

First, it will review the ArcMap Table of Contents. If the data exists, the code will remove it. Next, it will attempt to delete the original data source files and the data folder (this includes the CSV files and Shapefile). However, if a "lock" file is identified within the folder, the code will notify the user with an error message specifying that a Shapefile lock exists (see **Appendix – Figure 25**). When the user clicks the OK button to that error message, the code will then attempt to automatically remove the "lock" before deleting all remaining files associated with that data. The code is designed to attempt to release the "lock" twice. If, for some reason, the "lock" cannot be released, the code will notify the user that it is persistent and to manually investigate the situation. I was unable to create this situation during testing, but it is included in order to prevent the code from looping endlessly if a "lock" is not disappearing. Additionally, the code is designed to identify if multiple "locks" exist within one folder location. Although this would be a rare occasion, testing has shown this to occur when a user imports multiple Shapefiles into ArcMap from the same data hazard folder. If multiple "locks" are observed within the same folder, the code will notify the user of this situation and exit operation (see **Appendix – Figure 26**).

If a persistent "lock" or multiple "locks" are identified, the user will be required to manually investigate the situation. To assist with these rare occurrences, the Add-In toolset includes the Lock Release Button.

Operation-specifics for the Lock Release Button:

When a user is required to investigate a persistent Shapefile “lock” file or multiple “lock” files, the Lock Release Button is there to assist. When a user manually removes a Shapefile from the ArcMap Table of Contents, the Shapefile “lock” may never go away. The most notorious method of removal is to terminate the ArcMap session itself, or terminate the ArcGIS Cache Manager within Windows Task Manager. Neither of these options are ideal. It would be preferable to remove the Shapefile “lock” without interrupting the user’s ArcMap session. The Lock Release Button allows for this removal without disrupting the session.

After a user has removed a Shapefile from the ArcMap Table of Contents, this button can be clicked to cause the “lock” file to disappear. Once the “lock” is released, the user can verify that the “lock” is gone and attempt to re-download the data that previously encountered problems.

Interestingly enough, there are several ways to release these “lock” files without starting your ArcMap session. Utilizing the Runtime.InteropServices.Marshal.ReleaseComObject() is the ideal way of releasing “locks” within ArcObjects. However, this proved problematic for my code (to be expanded upon in the Discussion section). In the end, simply having the code change the ArcMap view from Data View to Layout View, then from Layout View back to Data View, consistently released any persistent Shapefile “locks”.

Operation-specifics for the Symbology Button:

The third and final button that is included within the Add-In Toolset includes the Symbology Button. This button exists for those situations where the user did not have the code automatically import/symbolize the Shapefile within ArcMap via the “Add to ArcMap” checkbox. Without this Symbology Button, the user would need to manually symbolize any Shapefiles added to ArcMap.

This button allows the user the freedom to add the Shapefile to ArcMap whenever they deem appropriate and symbolize it when/if they desire. If the user does want to symbolize a natural hazard shapefile that has been manually imported, the user is required to select the natural hazard data to be symbolized within the Table of Contents. Once the layer is selected, the user can click the Symbology Button to assign the appropriate values.

The code is programmed to identify layer names based on data source naming conventions. For example, if a user imports “noaa_2016_torn_20170810_checked.shp” into ArcMap and attempts to symbolize that layer, the code will run through a series of checks to verify that the data matches one of the symbology choices (earthquake, hail, hurricane, or tornado). First, the code will check to ensure that a layer exists within ArcMap. If no layers have been added, the Symbology Button will be grayed out and unavailable. It is designed to only be accessible while layers exist within the ArcMap session. Next, if layers do exist, the code will check to see if a layer has been selected within the ArcMap Table of Contents. If no layer is selected, or if multiple layers are selected, an error message will display explaining the problem and instructing the user to try again (see **Appendix – Figure 27**). Additionally, if a user does not select a Feature Layer (e.g. if the user selects a Basemap or Stand-alone Table), the code will present that same error message. Only Feature Layers can be considered for use with this Symbology Button. Once a Feature Layer has been selected, the code will run through additional tests to ensure that the selected Feature Layer is a Shapefile. If the Feature Layer is a Shapefile, the code determines if the Shapefile geometry consists of points or polylines. If the Shapefile consists of neither, then an error message will notify the user and exit. For the Shapefiles with point geometry, the code then attempts to identify the point data source. For those data sources that contain “usgs”, “hail”, or “torn” in the file name, the code will then verify that the headers match within those Shapefiles before proceeding to symbolize them accordingly. The same process occurs with the Shapefiles containing polyline geometry. The polyline Shapefiles are designated for Hurricane Shapefiles with “ibtracs” in the data source naming convention. If the code is unable to identify any of these naming conventions within the data source naming convention, an error message will inform the user that no layers match the symbolization options.

One feature of this workflow I have integrated involves the layer naming convention within the Table of Contents. The code has been designed to only look at the data source naming convention for “usgs”, “hail”, “ibtracs”, and “torn” keywords. The code doesn’t not refer to the Feature Layer’s naming convention within the ArcMap Table of Contents. The user has the flexibility to rename the Feature Layers within ArcMap and the code will still function properly for symbology purposes. The only caveat to this workflow is that the user must not change the data source naming convention to the Shapefile itself.

This concludes the functionality of the Add-In Toolset.

Results & Discussion

During the creation of this Add-In Toolset, numerous challenges were encountered and adjustments were made. Although I would have preferred to spend more than approximately eight weeks on this project, the results far exceeded my personal expectations, given the time constraints. The original scope of the project encompassed only one Add-In Button (the Hazard

Data Button), but I was able to modify the workflow and incorporate two additional Add-In Buttons to the Toolset before the deadline approached.

In retrospect, there are a few options within this code that I would have enjoyed investigating further. Most notably, the Shapefile “lock” issues should have been removable via ArcObjects with the Runtime.InteropServices.Marshal.ReleaseComObject() process. This was unsuccessful with my testing, however, and ended up being shelved. The COM Objects were related to workspaces within my ArcObjects coding, but something would continually remain unreleased.

An additional issue that presented problems for my code’s functionality had to do with the Hail and Tornado data. The raw CSV files downloaded from the web included a FromXY (slon/slat) field and a ToXY (elon/elat) field. These fields were meant to allow the user the ability to create line segments with the CSV files instead of data points. However, the accuracy of these ToXY fields was questionable in many cases. More recent CSV files contained ToXY values, but the older CSV files would have 0,0 populated for the ToXY values. When the code would attempt to draw line segments based on the FromXY values to the 0,0 values in the ToXY fields, the geoprocessing tool would attempt to draw the line segments from the United States to 0,0 Lat/Long near Africa. The size and length of the line segments would overload the ArcMap session and cause the system to crash. Numerous attempts to manipulate the CSV file yielded mediocre results. If I opted to remove the ToXY values of 0,0 from the CSV file, in some cases, that would remove 100% of the data. To alleviate the inconsistencies with the ToXY fields, I felt it was necessary to exclude them from the project altogether.

Throughout the course of the project, classmates provided useful feedback to consider enhancing one’s project capabilities. One classmate suggested a terrific idea where I include a feature in the code that would automatically update the latest year available as time progressed. The idea being that this would keep the toolset viable beyond the current timespan availability. As a result, the code now has a snippet that designates the latest year available as “Current year – 1”. Since 2016 data is the latest available data in 2017, when 2018 arrives, 2017 data should become available and that year will update/populate within the Timespan drop-down lists[9].

Conclusions and Future Study

With the creation of this Add-In Toolset, the user has the ability to easily access natural hazard data from the web, export the data into Shapefile format, and load/symbolize the data into ArcMap. These tool options allow the user to focus more on data analysis and less time on data preparation. With careful enhancements, the code could be enhanced to include many other natural hazard data options (if URLs for them exist). More functionality could be incorporated to allow the user to select natural hazards within specific months instead of years. The possibilities are numerous, and that is the beauty of programming: The user-friendliness can always be improved.

[9] Since this is an assumption and not a guarantee, the code will still need to be reviewed as each new year arrives. This should be considered in case NOAA or the USGS do not update the data URLs in a timely fashion.

References

Amirian, P. (2013). *Beginning ArcGIS for Desktop Development Using .NET*. West Sussex, UK: John Wiley & Sons, Ltd.

Boehm, A. (2016). *Murach's Visual Basic 2015*. Fresno, CA: Mike Murach & Associates, Inc.

Burke, Robert. (2003). *Getting to Know ArcObjects: Programming ArcGIS with VBA*. Redlands, CA: Esri Press.

Chang, K.T. (2008). Programming ArcObjects with VBA: A Task Oriented Approach. Boca Raton, FL: Taylor & Francis Group, LLC.

Nasser, H. (2015). *ArcGIS By Example*. Birmingham, UK: Packt Publishing, Ltd.

Ottemaller, Lars. (2014, October). The SEISAN Earthquake Analysis Software. *Incorporated Research Institutions for Seismology*. Retrieved from: https://www.iris.edu/hq/webinar/2014/10/the_seisan_earthquake_analysis_software

Williams, Weskamm, Baaser, Hinzen, & Bareth. (2008, July). SeismoGIS: A Tool for the Visualization of Earthquake Data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B8. Beijing 2008.* Retrieved from: http://www.isprs.org/proceedings/XXXVII/congress/8_pdf/12_WG-VIII-12/11.pdf

Zeiler, M. (2001). *Exploring ArcObjects: Vol. 1 – Applications and Cartography*. Redlands, CA: Esri Press.

Appendix

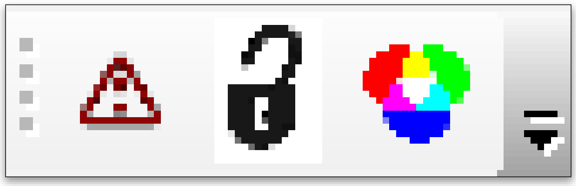


Figure 1

Figure 2

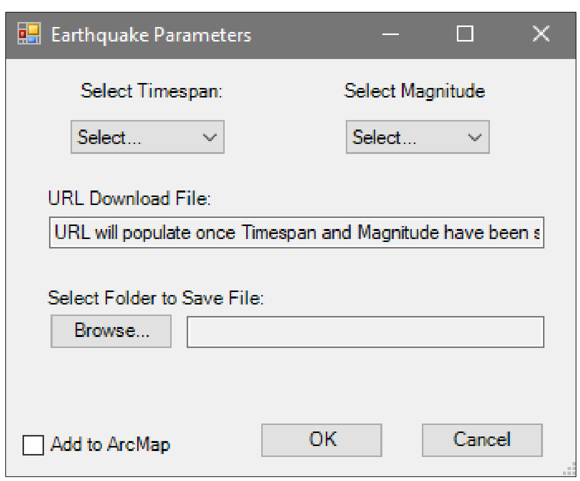


Figure 3

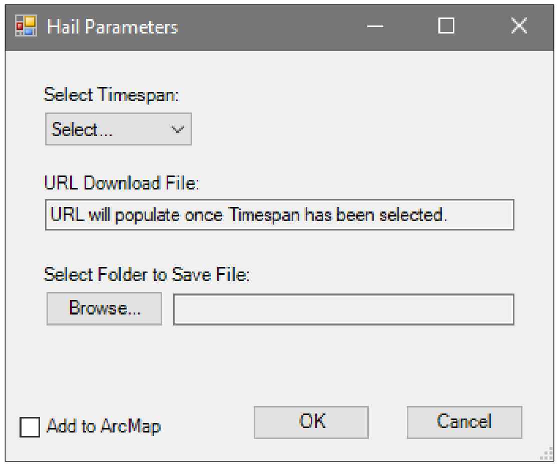


Figure 4

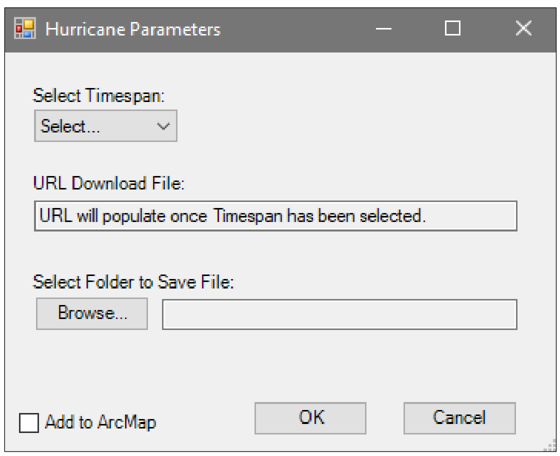


Figure 5

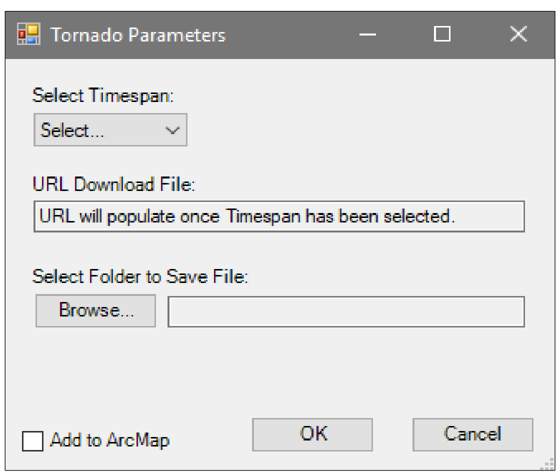


Figure 6

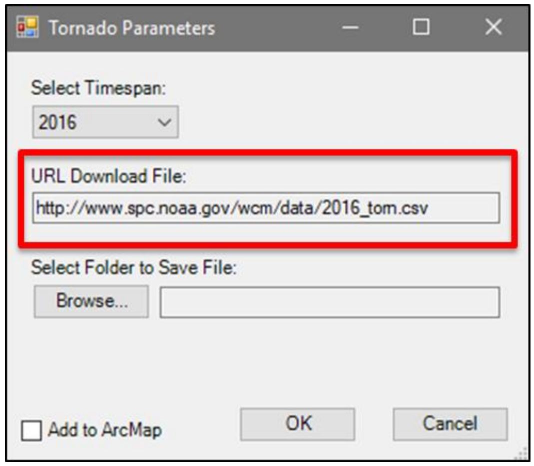


Figure 7

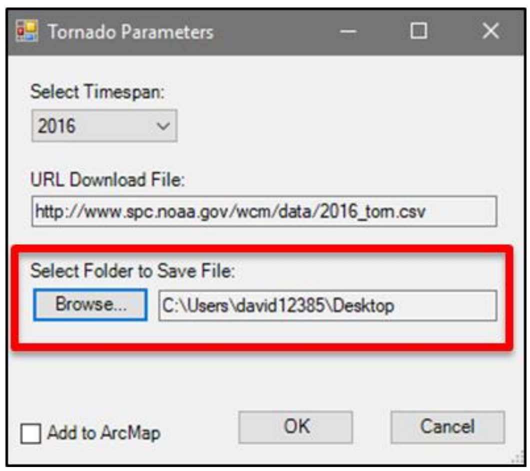


Figure 8

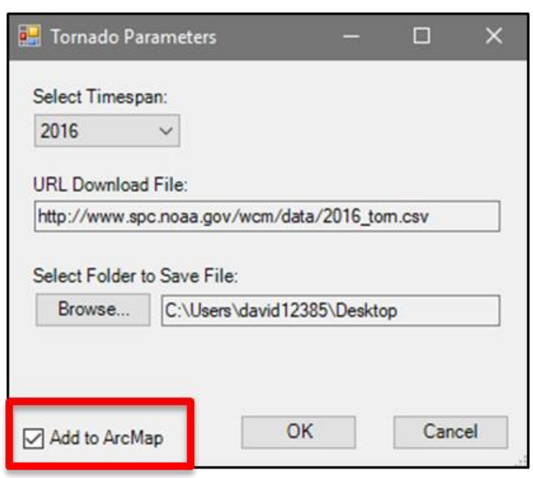


Figure 9

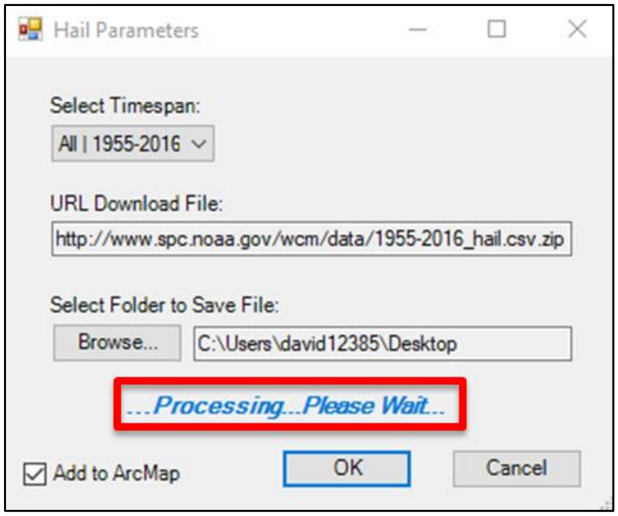


Figure 10

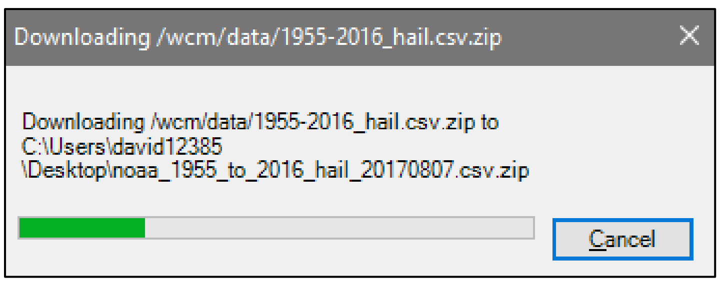


Figure 11

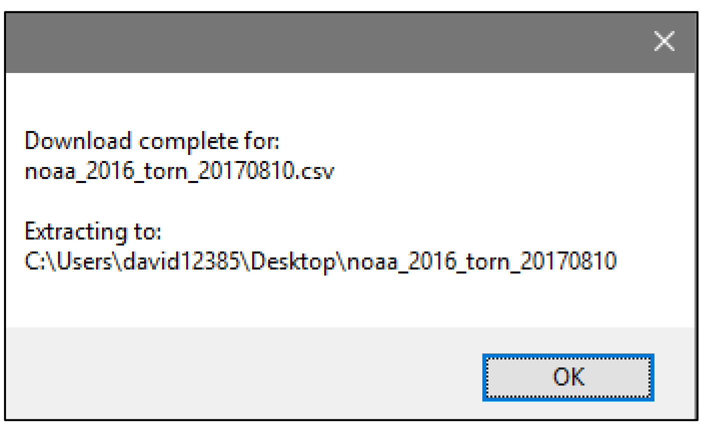


Figure 12

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
1	om	yr	mo	dy	date	time	tz	st	stf	stn	mag	inj	fat	loss	closs	slat	slon	elat	elon	len	wid	ns	sn	sg	f1	f2	f3	f4
2	1	1955	1	17	1/17/1955	16:39:00	3	TX	48	1	0.75	0	0	0	0	32.2	-101.5	0	0	0	0	0	0	0	227	0	0	0
3	2	1955	2	1	2/1/1955	10:30:00	3	MO	29	1	0.75	0	0	0	0	38.58	-92.58	0	0	0	0	0	0	0	135	0	0	0
4	3	1955	3	3	3/3/1955	13:02:00	3	IL	17	1	1.5	0	0	0	0	41.2	-89.68	0	0	0	0	0	0	0	175	0	0	0
5	4	1955	3	3	3/3/1955	14:00:00	3	IN	18	1	0.75	0	0	0	0	39.28	-87.4	0	0	0	0	0	0	0	167	0	0	0
6	5	1955	3	3	3/3/1955	16:00:00	3	IL	17	2	2.5	0	0	0	0	41.78	-87.78	0	0	0	0	0	0	0	31	0	0	0
7	6	1955	3	3	3/3/1955	19:45:00	3	IL	17	3	2.5	0	0	0	0	39.5	-90.08	0	0	0	0	0	0	0	117	0	0	0
8	7	1955	3	4	3/4/1955	15:00:00	3	IL	17	4	2.5	0	0	0	0	38.38	-88	0	0	0	0	0	0	0	47	0	0	0
9	8	1955	3	4	3/4/1955	16:30:00	3	KY	21	1	1	0	0	0	0	37.88	-86.08	0	0	0	0	0	0	0	163	0	0	0
10	9	1955	3	4	3/4/1955	17:28:00	3	KY	21	2	0.75	0	0	0	0	37.88	-86	0	0	0	0	0	0	0	163	0	0	0

Figure 13

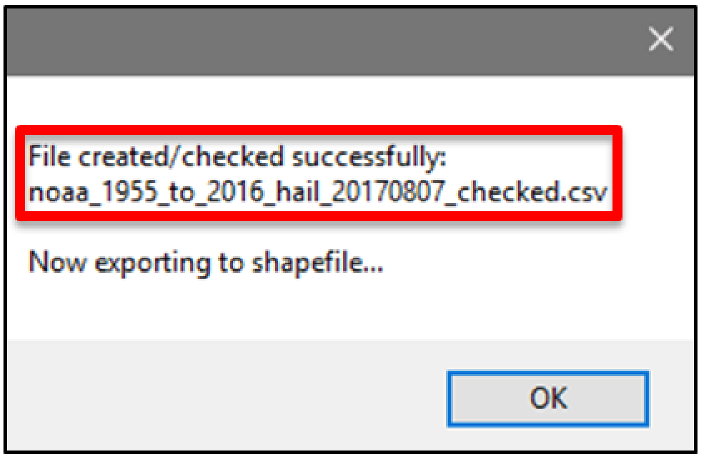


Figure 14

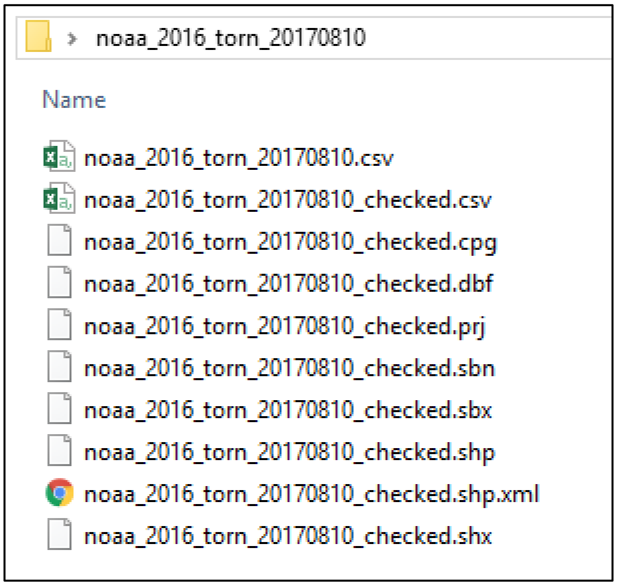


Figure 15

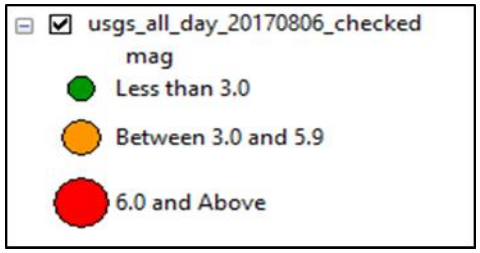


Figure 16

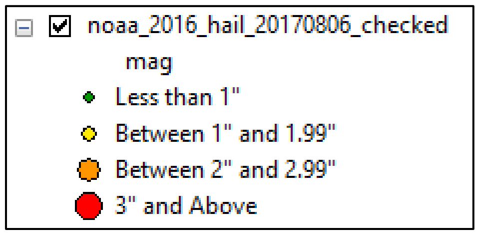


Figure 17

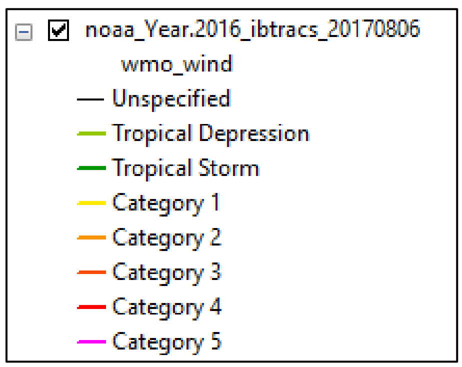


Figure 18

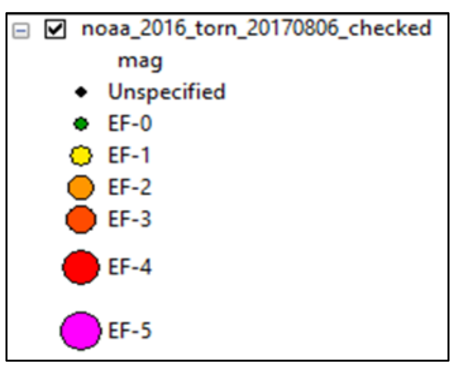


Figure 19

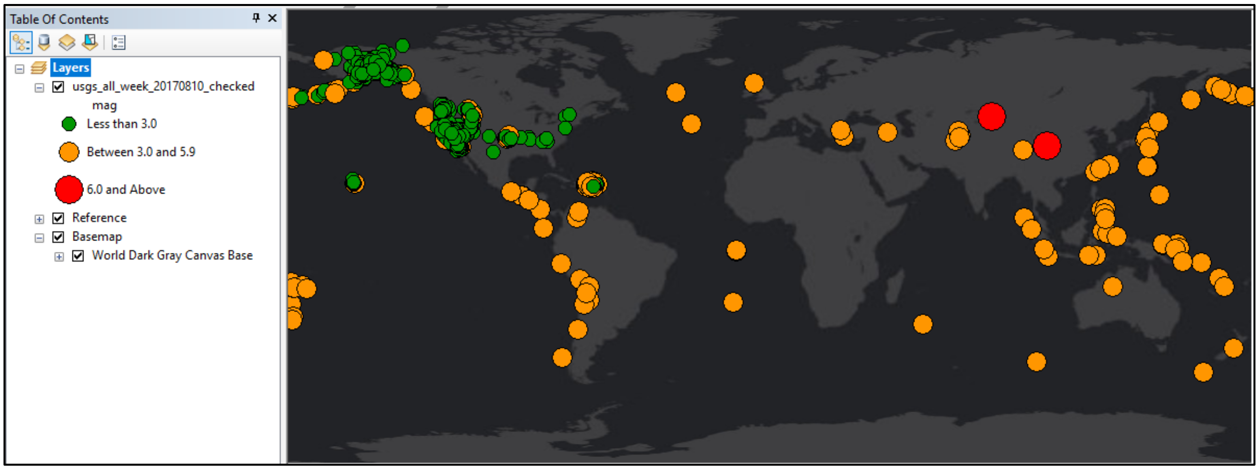


Figure 20

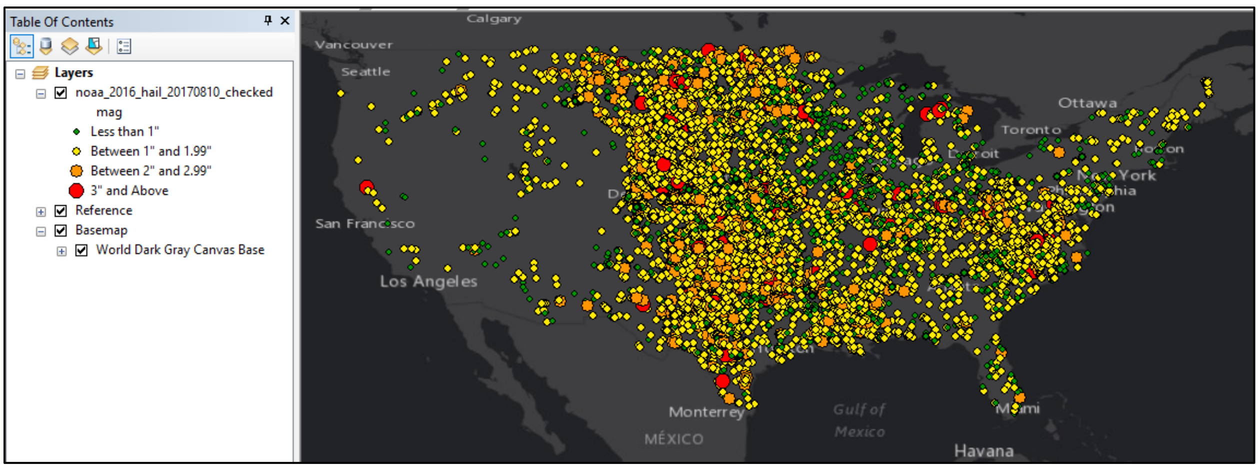


Figure 21

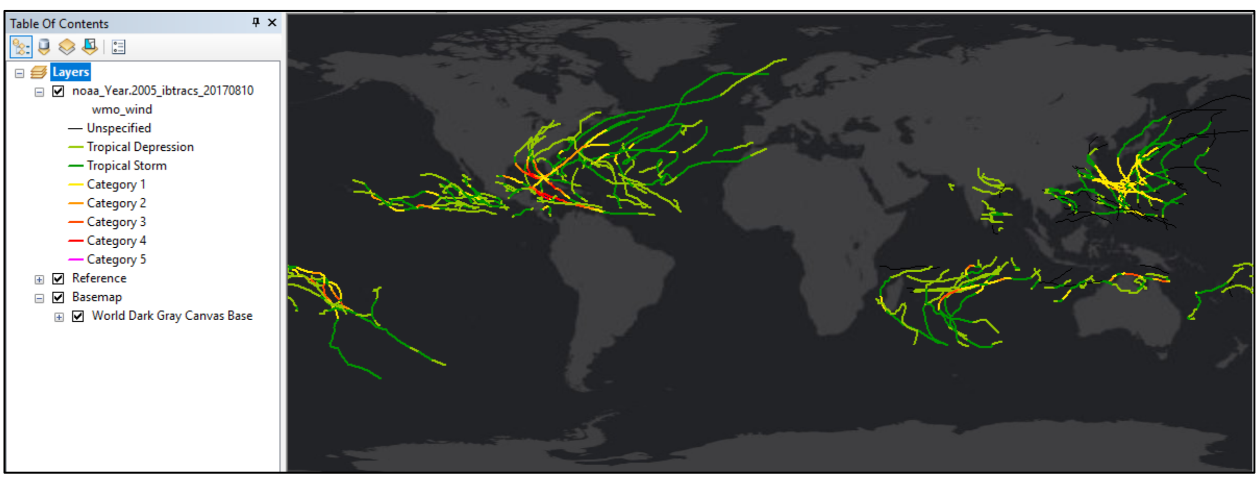


Figure 22

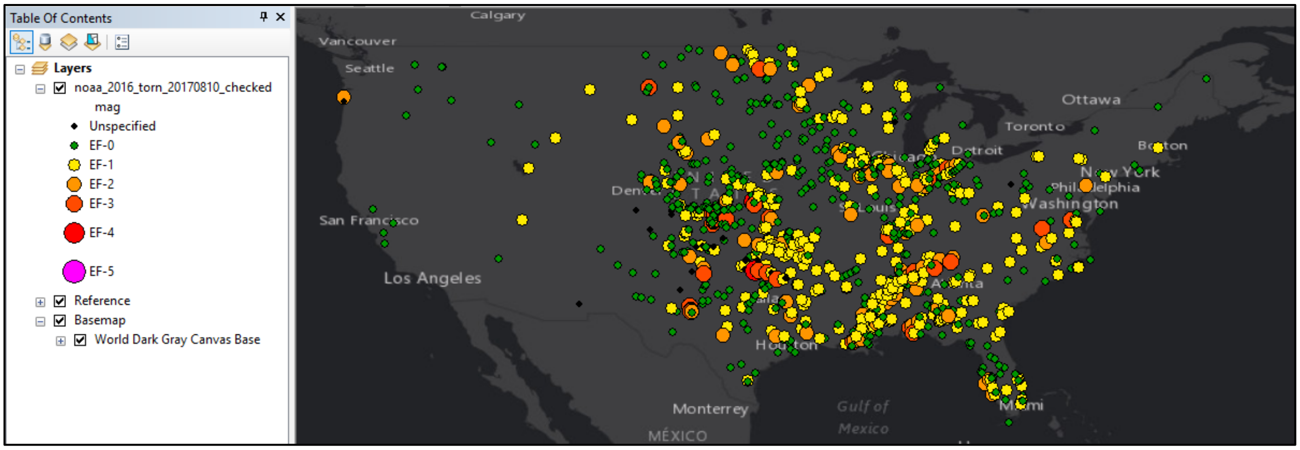


Figure 23

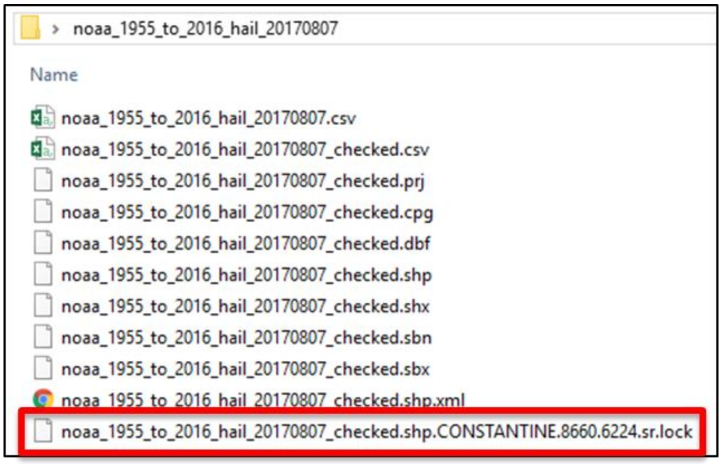


Figure 24

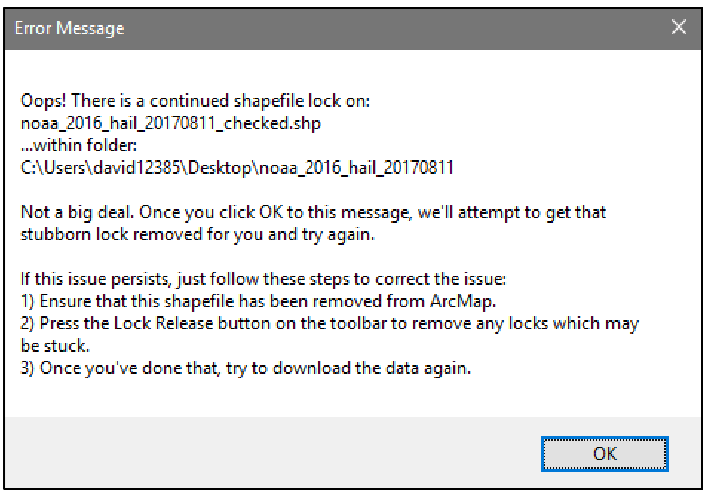


Figure 25

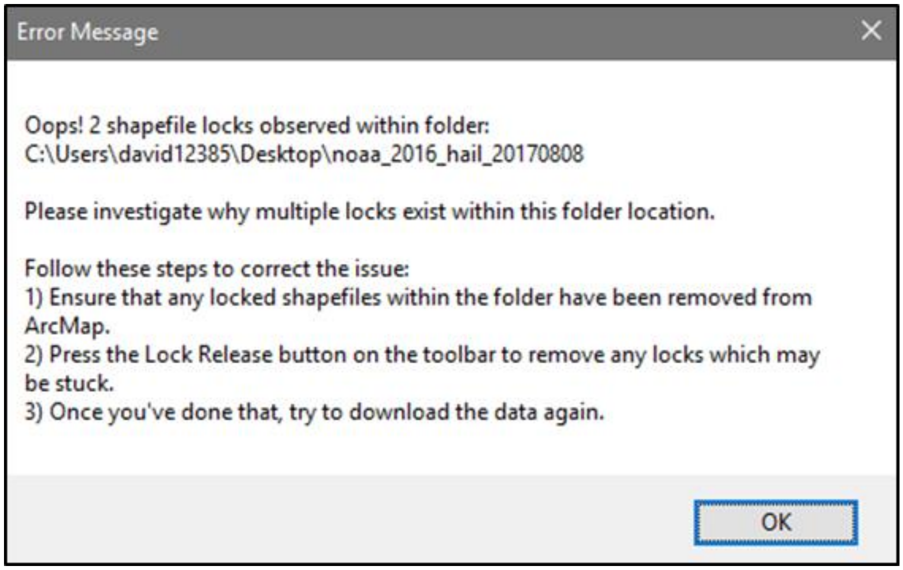


Figure 26

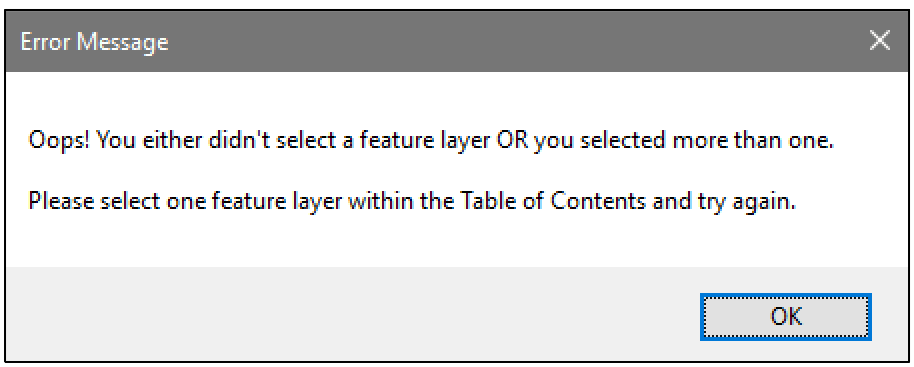


Figure 27