

# **Automated Conversion and Processing of Weather Data in Near-Real Time**

Daniel Konde – QSS Group Inc.

Rory Moore – Raytheon

## **1.0 Abstract**

In the spring of 2003 the National Weather Service (NWS) developed an ArcIMS™-enabled Web site to display hurricane and related weather data. This site was developed to provide emergency managers with a tool to assist with hurricane preparedness planning. Since weather data is dynamic, it was imperative for the site to create and/or update data layers in near real time. This paper describes the methods and techniques used to automate data creation, update and processing functions. This paper focuses on methods for automating data extraction and data processing functions, and provides lessons learned during the development process.

## **2.0 Background**

In August 2002 a team was commissioned by the NWS Science and Technology Committee to investigate the feasibility of using Internet Map Services as the foundation for integrating and displaying hydrometeorological data via the Internet. In the team's April 2003 report, it was recommended that a prototype application be developed based on ESRI's ArcIMS™ software. The Science and Technology Committee accepted the report and agreed to fund the building of a prototype, which is the subject of this paper. The Web site was given the designation EMHURR.

## **3.0 Overview**

A unique challenge to the development of EMHURR was the acquisition and processing of the data sets served by the application. Because the majority of EMHURR data sets are dynamic (constantly changing), an automated system for processing source data sets into an ArcIMS™-compatible format (shapefile) in near real time was required. The automated processing tasks can be categorized as data extraction tasks, or data processing tasks. Extraction tasks involved processes for downloading, parsing, and decoding source data sets. Processing tasks involved methods for creating shapefiles from source data sets or updating existing shapefiles.

The source data used by EMHURR was obtained in multiple formats from multiple NOAA line offices and NWS program offices. Source data formats included HTML, XML, text, GRIB1 and GRIB2. Each of these data formats was used to either create shapefiles or update shapefiles being served via EMHURR.

The majority of processing tasks were accomplished using Visual Basic, Avenue, and SoftTree's 24 x 7 Scheduler software. In the following paper, data extraction and data

processing tasks are described for each of the data layers included in the EMHURR application: Storm Track, Watch Warning and Advisory (WWA), Flood Outlook, River Conditions, Wind Forecast, and Precipitation Forecast.

## 4.0 Storm Track

The storm layer was composed of a point shapefile representing forecast locations, a line shapefile representing the storm path, and a polygon shapefile representing the cone of uncertainty. The point forecast layer displayed forecast points at 12-hour intervals to a forecast lead time of 120 hours or 5 days. The source data was obtained from the National Hurricane Center in an HTML format.

### 4.1 Data Extraction

The storm HTML files were downloaded, and parsed using a Visual Basic (VB) script that was triggered by the SoftTree 24 x 7 Scheduler software. The VB script parsed the storm HTML files, produced output storm arrays, and created output storm text files from the storm arrays. A text file was created for the Atlantic, Eastern Pacific, Central Pacific, and Western Pacific regions. These text files were subsequently used to create storm shapefiles for each region.

Initially the VB script had a problem interpreting end-of-file (EOF) characters in the storm HTML files, which caused some characters to be misread or omitted. This problem was resolved by creating a dump file (see Section 10.0, Lessons Learned). The dump file enabled the storm HTML files to be read into the VB script byte by byte. The script then converted the bytes to characters and stored the values in the storm arrays. The storm arrays were created using unique parsing routines for each region. The arrays were then used to create text files for each region. Figure 4.1 provides an example of a storm text file created using this process.

```
# Message Type|Message Time|Type|Name|Advisory|Move Dir|Move Speed|Min Pressure
# Time|Lat|Lon|Max Wind|Gust
#
TCMAT4|JUL 14 2003 0900|TROPICAL STORM|CLAUDETTE|23|320|5|992
JUL 14 0900|26.3|-92.5|55|65
JUL 14 1800|26.8|-93.1|55|65
JUL 15 0600|27.2|-94.3|60|75
JUL 15 1800|27.5|-95.7|65|80
JUL 16 0600|27.7|-97.3|65|80
JUL 17 0600|28.2|-99.7|25|35
JUL 18 0600|29.0|-102.4|15|20
```

Figure 4.1 – Storm Text File

### 4.2 Data Processing

The forecast points shapefile was created from longitude and latitude values extracted from the storm text file. These values were used to create an event theme, which was

then converted to the forecast points shapefile. The storm path was created by drawing a line between the forecast points. The storm track layer comprises both the forecast point and storm path shapefiles. The cone of uncertainty polygon shapefile was created from the forecast point shapefile. A circle was drawn around each forecast point using a constant defined by the National Hurricane Center. The circles represent the likelihood (based on historic error measurements) a forecast point will be located within the circle. The circles increase in size with each 12 hour forecast point. The circles were then connected using left and right intersect points to create a polygon cone of uncertainty. These data processing steps were automated with Avenue scripts.

## **5.0 Watch Warning and Advisory (WWA)**

The WWA layer showed county shapefiles whose attributes included severe weather alert events. This layer could account for as many as ten weather warnings, watches or advisories (per county) at a particular time. The Watch Warning and Advisory Events included severe thunderstorms, tornados and other severe weather alerts. The WWA is a county polygon shapefile for the entire United States. This file was updated every six minutes using SoftTree's 24 x 7 Scheduler.

### *5.1 Data Extraction*

The WWA layer's original input files were XML files. The method for reading these data files into Visual Basic modules was the same as the method used to read in the HTML files for the Storm Track's HTML files. The same erroneous EOF characters found when reading HTML files arose when downloading WWA XML data files (see Section 10.0, Lessons Learned). As with the hurricane HTML files, the WWA XML files were read in byte by byte, converted to characters, and stored in a string array. Visual Basic functions were written to parse the XML file according to XML tags. The relevant data were extracted, and used to create an output array called "XML array."

### *5.3 Data Processing*

The WWA shapefile attribute table contained ten fields, each representing a type of severe weather event. Each event had an expired time, which indicated when the event would no longer be valid. Each U.S. county in the table could potentially have multiple events occurring at any given time. The range of possible events for each county ran from zero to ten. If a county had the maximum number of events (10), then each time the dbf table was updated (every few minutes), it required that the 10 most recent (latest expire time) events be captured. If an event's expiration time had passed, that event would be removed from the table. If an event was still "active," it would continue to be displayed in the table.

The event evaluation logic and the dbf table read and update operations were all performed in Visual Basic 6.0. Specific information from the dbf table header was read and held in memory. The information was used to determine the structure of the dbf

table. The dbf table was then converted to a string array, called “dbf array.” The FIPS code in the string array was associated with the corresponding FIPS code in the XML array. If a match of county FIPS codes was found, the dbf array was updated with the value contained within the XML array. The updated dbf array was then used to update the attribute table in the WWA shapefile.

## 6.0 Flood Outlook

Flood Outlook is a polygon data layer that delineates areas where flooding is likely to occur. The source data for the flood outlook shapefile was a delimited text file obtained from the Hydro-Meteorological Prediction Center.

### 6.1 Data Extraction

Flood Outlook comma delimited text files were downloaded and decoded using a VB script triggered with the SoftTree 24 x 7 Scheduler software. The VB script parsed the flood outlook text file to eliminate unnecessary header information, and expired flood activity. The script also created a flood outlook array, and from that array created a new flood outlook text file.

The new flood outlook text file contained the shape count (number of shapes), number of points per shape, date range and latitude/longitude pairs for each shape. Figure 6.1 displays an example of the flood outlook text file.

```
Shape Count = 2
7 2 JUN 7 - 12, 2003
1, -79.67, 33.66
2, -80.04, 33.82
3, -80.39, 33.74
4, -80.45, 33.41
5, -80.17, 33.09
6, -79.63, 32.98
7, -79.45, 33.28
```

*Figure 6.1 – Flood OutlookText File*

### 6.2 Data Processing

The new text file was parsed, and date, longitude, and latitude values were written to a string array. The sting array was used to create a set of virtual points. These virtual points were then added to an ArcObject array of points called a point collection. The point collection was set to a virtual polygon, and the new polygon was closed. Simultaneously a new empty polygon feature class was created. The feature class was set to a shapefile, and a new empty feature (record) was added to the shapefile. The first virtual polygon defined in the point collection array was then applied to the geometry attribute of the first feature (record) in the shapefile feature class. Next, string attribute fields were defined for the feature class and were populated with the date, latitude, and

longitude values specified in the initial string array. This process was repeated in a nested loop until each feature defined in the point collection, and its corresponding attributes were added to the shapefile. This process was automated using Visual Basic with ArcObjects.

## **7.0 River Conditions**

There were two river condition layers included with EMHURR. The Crest Forecast layer displayed only river stations with flood stage values forecast to exceed flood stage. The Current Stage layer displayed all river stations regardless of stage value. The river conditions data were obtained in a delimited text format from the Lower Mississippi River Forecast Center (RFC) and West Gulf RFC.

### *7.1 Data Extraction*

A VB script was used to download and decode the river conditions text file. The text file contained values required for updating the stage value, crest value, and site ID fields in the river conditions shapefile attribute table (dbf). The text file was then written to a river array.

### *7.2 Data Processing*

In order to determine which records in the river conditions attribute (dbf) table needed updating, the river conditions dbf file was written to a string array called “dbf array.” The river array was compared to the dbf array using the site ID value. Each ID value in the river array was then matched with the corresponding Site ID value in the dbf array. The dbf records that matched were then updated with the stage and crest values from the river array. A new river conditions dbf file was created from the updated dbf array. This file was then copied to the river conditions data directory on the ArcIMS™ server, effectively replacing the original river conditions shapefile attribute table. This processing was automated using Visual Basic.

## **8.0 Wind Forecast**

Both 12- and 24-hour wind forecast layers were viewable through EMHURR. The layers are composed of polygons rendered by tropical depression, tropical storm, or hurricane wind speeds. The source of these data was the National Digital Forecast Database (NDFD) maintained by the Meteorological Development Lab (MDL).

### *8.1 Data Extraction*

GRIB2 is a compressed binary data format developed by the NWS. The data was accessed via FTP download, and converted to a floating-point (FLT) file using a

conversion utility developed by the NWS Meteorological Development Lab. This utility can be accessed at the following URL:

<http://www.nws.noaa.gov/tdl/iwt/grib2/decoder.htm>.

The GRIB2 data were downloaded hourly using a VB script triggered by the SoftTree 24 x 7 Scheduler software.

## *8.2 Data Processing*

The FLT files contained wind forecast values for a 3- hour time period. Therefore, four FLT files were required to create a 12-hour forecast, and eight FLT files were required to create a 24-hour forecast. The FLT files were converted to ESRI GRIDS and manipulated using a variety of raster processing functions. The first function combined layers using a max function to create a max wind GRID for 12 hours and 24 hours, respectively. Next, the max wind GRIDS were reclassified to create 12- and 24-hour storm GRIDS. The storm GRIDS contained three values that corresponded with tropical depression, tropical storm, and hurricane wind speeds. The storm GRIDS were then converted to polygon shapefiles. The output storm shapefile contained hundreds of multipart polygon records for each storm category. To accelerate display speed in ArcIMS™, the storm shapefile was dissolved using the storm value. The output shapefile contained one record for each storm category. The shapefile was then pushed to the data directory on the ArcIMS™ server. These processing steps were automated using Visual Basic with ArcObjects. Raster processing functions required a Spatial Analyst license.

## **9.0 Precipitation Forecasts**

One- and five-day Quantitative Precipitation Forecast (QPF) data layers were included with EMHURR. These layers were composed of polygon features rendered by numeric ranges based on precipitation values. The source of these data was the National Digital Forecast Database (NDFD) maintained by the Meteorological Development Lab (MDL).

### *9.1 Data Extraction*

The extraction process used for the QPF data was identical to the process used for the wind forecast data. The data was accessed via FTP download and converted to a FLT file using a conversion utility developed by the NWS Meteorological Development Lab. This utility can be accessed at the following URL:

<http://www.nws.noaa.gov/tdl/iwt/grib2/decoder.htm>.

The GRIB1 data were downloaded once a day using a VB script triggered by the SoftTree 24 x 7 Scheduler software.

## *9.2 Data Processing*

The QPF data was processed in a manner similar to the Wind Forecast data. The FLT files contained precipitation forecast values for a 24-hour time period. Therefore, three FLT files were required to create the 3-day forecast, and five FLT files were required to create a 5-day forecast. The FLT files were then converted to ESRI GRIDS. The GRIDS were combined together using an add function to create cumulative precipitation GRIDS for 3- and 5-day periods. The precipitation GRIDS were then converted to polygon shapefiles and pushed to the data directory on the ArcIMS™ server. These processing steps were automated using Visual Basic with ArcObjects. Raster processing functions required a Spatial Analyst license.

## **10.0 Lesson Learned**

During the development process file reading, data conversion, license verification, and spatial referencing issues were encountered. Each of these issues required additional research & collaboration to solve. In this section these development challenges will be cited, and the work around solutions described.

### *10.1 File Reading*

The Storm Track HTML file, and WWA XML files were downloaded using an ftp process. When the files were read into the VB parsing script, the script failed. The cause of the problem was discovered while using a “dump” utility on the input files, and visually inspecting them for any anomalies. A “dump” utility takes a file as its argument (in our case an HTML or XML file) and returns an output of the file’s ASCII characters along with each character’s numerical representation. It was discovered that the process of downloading the files from the web caused some issues relating to Visual Basics’ handling of EOF (end of file) characters. Thus, the VB parsing script was not able to correctly read the data contained in the input files. Inside a computer, ASCII characters are represented by numbers. The numerical representation of each character allows one to identify a character in an ASCII table. The “dump” utility we employed output each character in its hexadecimal form. Hexadecimal form means that each character was displayed in its base-16 number representation. For ordinary characters, used in everyday language, the ASCII hexadecimal representation is trivial. The dump utility allowed us to identify any unseen characters (i.e. carriage returns, linefeeds, etc.) and replace them during file read operations. This required the VB code to be modified to read the HTML and XML files in binary mode (byte by byte) as opposed to line by line. In doing so, we were able to identify the problem character sequence, replace the sequence during the file read operation, and correct the early EOF situation.

### *10.2 Data Conversion*

The QPF and Wind Forecast data layers were downloaded in GRIB formats. They were converted to flt files, and the flt files were then converted to ESRI GRIDS. During the flt

to GRID conversion the VB script continued to crash. Eventually with ESRI's support it was determined that the output GRID file names needed to be limited to seven characters or less. File names greater than seven characters caused the script to crash. These naming convention constraints appear to be limited to file names defined in code. The same constraints do not exist when manually processing through ArcToolbox.

### *10.3 License Verification*

The OPF and Wind Forecast data layers required the use of raster overlay functions. When the VB processing script called these functions the script failed. A Spatial Analyst license was required to utilize these raster functions. A current Spatial Analyst license was present on the server, but the license was not recognized when the function was called. To solve this problem VB code was added to the script, preceding each function to verify the existence of the Spatial Analyst license. Once the license was verified, the functions were executed, and the script successfully ran.

### *10.4 Spatial Referencing*

In order to perform overlay functions on ESRI GRIDS, the GRIDS are required to share the same spatial reference. When the VB processing script attempted to perform a raster overlay function the script failed, and returned a spatial reference error. The input GRIDS were already in the same projection, yet the error still occurred. To solve this problem VB code was added to the script to set the spatial reference of the input GRIDS to match each other. Once the license was verified, the functions were executed, and the script successfully ran.

## **11.0 Conclusion**

This paper has focused on the methods used to automate the decoding, and processing of weather datasets utilized in the EMHURR ArcIMS application. Data was accessed from multiple NWS and NOAA sources in a variety of data formats. The source data was decoded, and processed in a near real-time fashion using Avenue, Visual Basic and ArcObjects. The SofTtree 24 X 7 scheduler software was used to execute these scripts at predefined data update intervals. Currently EMHURR is going through phase II development in preparation for the 2004 hurricane season. The most significant enhancement to EMHURR is the migration to a database centric architecture. EMHURR now accesses ArcSDE layers instead of shapefiles. This architecture is improving the speed, scalability, and capacity of the site.



## **12.0 Acknowledgements**

The authors would like to acknowledge the following key contributors to the development of EMHURR: Ira Graffman (NWS/Office of Science and Technology, Systems Engineering Center), for his technical leadership, John Kozimor (QSS Group Inc.), for ArcIMS development, Melody Magnus (Raytheon Corp.) for creating the legend graphics, Dario Leonardo (NWS/Office of Science and Technology, Systems Engineering Center), for providing systems administration support, Rita Lewis (QSS Group Inc.) for her proof reading skills, David Welch (Lower Mississippi RFC), for creating and coordinating the river stage data, Frank Bell (West Gulf RFC) for providing precipitation data, Ralph Meiggs, and Marlene Patterson (National Environmental Satellite Data, and Information Service), for providing GOES satellite imagery.