

# Automated Approach of Cataloguing TeraScan Images Utilizing Active Server Pages

Garry Cameron, Kaiem Frink, Shaquera Mosley, Lee Smalls, Jr., Danielle Wright  
1704 Weeksville Road, Box 672  
Elizabeth City, North Carolina 27909

**Abstract** - The Center of Excellence in Remote Sensing Education and Research (CERSER) on the campus of Elizabeth City State University is currently tasked with the responsibility of receiving remotely sensed Advanced Very High Resolution Radiometer (AVHRR) and Sea Wide Field-of-View Sensor (SeaWiFS) data for research projects year round. This data is collected, processed, annotated, and transformed into images in the Tagged Image File Format (tiff). These tiff images are then uploaded to the CERSER archive library server located at <http://cerser.ecsu.edu>. Once uploaded, they are converted into various resolutions and their information is added to a tracking database maintained with Microsoft Access software. This database provides a searchable means for retrieving satellite image data through various parameters.

The automatic conversion of the original tiff file to various Joint Photographic Experts Group (jpeg) format sizes and the entry of information into the tracking database were completed during this project. Active Server Pages were developed utilizing Visual Basic Scripting, Structured Query Language, and Command Line instructions to the software package ImageMagick® to complete the tasks. The final file developed was added as a scheduled task to be implemented at a certain time each day on the CERSER platform. This file inventories the "original" folder located on the CERSER server to determine if any files have been uploaded and then processes those that have.

## I. INTRODUCTION

### A. Overview

The Center of Excellence in Remote Sensing Education and Research (CERSER) website currently maintains a database of images processed by the Terascan System. The NOAA or SeaWiFS satellites make multiple passes each day feeding data to the Terascan system in Dixon Hall. This set of data is then processed by the TeraScan system and an image in Tagged Image File Format (TIFF) is produced and placed on the CERSER server in a folder labeled "original". Each image once processed must then be renamed, resized, converted to Joint Photographic Experts Group (JPEG) format and added to a database. This process is done manually by an individual using a graphics software package and manually entering the image information into the database for each satellite.

This year's Multimedia team took on the task of automating this process utilizing Active Server Pages (ASP) and graphic software packages that can be controlled by the Visual Basic Scripting language. The team tried several graphic software packages such as Macromedia Fireworks and Adobe

Photoshop, settling on the open-license (free) package ImageMagick® due to its ability to be implemented from the Windows command line.

1. The outline of the steps that this process is:
2. Call the ASP page daily utilizing the Windows Scheduler
  - a. ASP code processes the image using the following steps:
  - b. Convert Date and Satellite information from the file name
  - c. Open Terascan Database to obtain new file name
  - d. Write new record to the database
  - e. Open ImageMagick®
  - f. Convert/Resize/Rename original image to Actual, Medium, Low, and thumbnail folders
  - g. Delete original image file

### B. Active Server Pages

Active Server Pages (ASP) are HTML pages that include one or more scripts that are processed on a Microsoft Web server and returned as HTML code to the user. Web pages that have an .asp extension (instead of an .html or .htm extension) are commonly used to retrieve information from a database. A user often makes a request of a particular type which is processed by the script before being returned to the user. ASP pages can be written in a variety of languages including Visual Basic Script (VBScript), JScript, PerlScript, and Javascript. The default language for the Microsoft Internet Information Services (IIS) Server is VBScript, but this can be changed by using the "@Language" command when writing the code. For this project, VBScript was utilized as the CERSER server is based on the IIS Server software.

### C. Databases

Images processed by the Terascan system in CERSER must be entered into a database for search and sorted retrieval. The information to be entered into the database is contained within the file name and must be converted to be placed into the correct locations.

A database is a table that gathers all types of information and places it into different categories. The physical equivalent of a database is a rolodex card system where each card is equal to a record in the database. The database is composed of fields which hold the same type of information for each record such as name, date, school, etc. The field name describes the

data contained in that cell. Field names that the TeraScan database utilizes are: ID, Date, Satellite, Product, Description, and Event. Once the field has been named, the type of information to be stored in the field must be determined. Some of the choices are date, number, text, currency, etc. Other parameters may include field size, required, default value, and others in order to make the data entered as structured as possible.

ID	Date	Satellite	Product	Description
1	9/16/2003	NOAA-17	0	Image of Hurricane Isabel impacting the NC
2	6/27/2003	SeaWifs/Orbview-2	0	Image From Tropical Storm Bill
3	6/27/2003	NOAA-15	0	Image From Tropical Storm Bill
4	6/28/2003	NOAA-15	0	Image From Tropical Storm Bill
5	6/28/2003	NOAA-16	0	Image From Tropical Storm Bill

Figure 01

The Multimedia Team will be retrieving and entering data from three fields: ID, Date, and Satellite. The original information for these fields can be found in the file title generated by the Terascan system. Other fields can be modified at a later time either through Active Server Pages or by accessing the database directly.

#### D. ImageMagick®

ImageMagick® is a software package used to create, edit, and create bitmap images. It can also read, convert and write images in a many formats. ImageMagick® is supplied with full source code that can be freely used, copied, modified and distributed. It supports over 90 image formats including the formats GIF, JPEG, JPEG-2000, PNG, PDF, PhotoCD, TIFF, and DPX. ImageMagick® can crop, change colors, apply various effects, and rotate images. It can also add text, lines, polygons, ellipses and curves to images.

ImageMagick® was chosen for its unique capability to be controlled by the Windows command line. This allowed the use of ASP pages to access the functions of resizing, conversion, and file renaming directly. A command line interface (CLI) is a method of working with a computer. Commands can be entered as lines of text from the keyboard and output is received back as a text line also. The command line interface was preceded by punch cards and followed by the graphical user interface.

Another advantage of ImageMagick® is that it can be freely used on as many computers as needed.

#### E. Scheduling

An operating system is software that manages the relationship between applications, the hardware that makes up a computer system, and the user of the system. The CERSER server is using Windows 2000 as an operating system which enables users to set up scheduled tasks.

Scheduled tasks allow users to have the operating system automatically run a program or open a certain file at a set time. The Multimedia Team is using the scheduled task program to execute an Active Server Page file that will process images from the CERSER TeraScan system to be added to the CERSER website.

Scheduled tasks are created using the “Scheduled Tasks Wizard”. The first step is to select Start>All Programs>Accessories>System Tools>Scheduled Tasks. This

opens the Scheduled Tasks window. Next select “Add Scheduled Task” by double clicking to start the wizard. Once the wizard is open, select “Next”. Now select the program to be scheduled. If the software desired is not available in the list, then select “Browse” and locate the program. Next name the task and select how often the scheduled task should run. On the next screen select the time the task will start, how often it will run, and the starting date. The next action is to enter a user name and password which is entered twice. The user name defaults to the current user, which in this case, is the Administrator. This completes the parameters needed to make a scheduled task. The last screen allows the option of opening the “advanced” properties for the task if needed.

Scheduled Tasks can be modified by double-clicking the created task. The “Advanced Schedule Options” appears allowing changes to be made to the properties of the task. These properties include the task start date, how often to run the task, number of times to run the task and other options included in the scheduler

## II. THE CODE

### A. Pseudocode

Before beginning the task of coding the ASP pages, Pseudocode was developed to allow the team to record how the structure of the code should function. Pseudocode is a simple method of describing a process before implementing the actual programming language. An example would be the output of a variable to the screen. In pseudocode this might read “Display on the screen the variable X”, while the actual coding would read “reponse.write (x)”.

### B. Beginning the ASP Code

The first step in writing the actual ASP code was to tell the server that ASP code starts. This is done with the symbols “<%” for start and “%>” to end the code. The ASP start is followed by an opening Hypertext Markup Language (HTML) comment, “<!--” which allows non ASP browsers to bypass the HTML code. This is followed by the declaration of the variables utilizing the “dim” command. The only variance in the project code was the use of “Public” for the variable “lastID” as it was used in multiple functions. This allowed the contents of the variable to be shared by multiple functions.

The last part of the opening code is to create an instance of the object “Scripting.FileSystemObject”. This object contains methods which perform on files such as copyfile, deletefile, read, and write files. The next line of code utilizes this object with the method “getfolder” to set the variable “fo” equal to the contents of the folder “original”. This is the location of the images processed by the TeraScan system and not uploaded to the CERSER website.

### C. Beginning the ASP Code

The command “For” begins the image processing loop. This code checks the contents of the “original” folder and will loop for each file in the folder. If now files are found, the code exits the loop, closes all variables out, and writes the update attempt time to the TeraScan database and to the CERSER server screen.

#### *D. Start Image Loop*

The command “For” begins the image-processing loop. This code checks the contents of the “original” folder and will loop for each file in the folder. If no files are found, the code exits the loop, closes all variables out, and writes the update attempt time to the TeraScan database and to the CERSER server screen.

#### *E. Date & Satellite Name Conversion*

This section of code utilizes the “Mid” command to extract characters from the file title to be converted to the proper date and satellite name format used by the Access database. The “Mid” syntax is mid(string, start, length) what utilizes the file name as the string.

After the date has been converted the code then begins to convert the satellite name presented in a shortened form from the file, which is being used and is converted to the format utilized by the TeraScan database.

#### *F. Database*

This section of the code sets the variable “dbpath” to the location of the database utilizing the “fs.getfolder” method. The path to the folder is obtained by using the server.mappath, which is then combined with a string. After being combined with the string and the filename, the path to the file is complete. After setting the path to the database the database is then opened. When the database is opened an instance of the object ADODB.Connection is made and then assigned to a variable. After being assigned to a variable a connection is then made to the database.

#### *G. Fields*

When opening and selecting an ID field an instance is made of ADODB.recordset is made and is assigned to the variable “rs”. After being assigned to the variable, the “open” method of the object “rs” retrieves a set of records from the database. The method that is used here uses the Structure Query Language (SQL) to then select the “ID” field from the “images” table and sorts them in descending order. After completion of this task the program then begins a “for” loop adding on the highest “ID” in the field. After finishing the first loop, the program then goes “next” and closes this field. After the loop ends the program now assigns values to the variables “description” “curevent” and “product” with “description” being set as null.

#### *H. Image Table*

The program now uses the “open” method of the object “rs” to open the “images” table in the terascan\_trial.mdb. The parameters of the “open” method are “Images” which is the name of the table to be opened, “1” sets restrictions on access to the database, “3” chooses that the database will be locked after the “update” command, and “2” states that a table will be opened.

#### *I. Adding a New Record*

Once the database is open, a new record is created utilizing the “AddNew” method of the object “rs”. Each field of the table is updated using the variables set since the code started. Once the data has been added to the table, the “rs.update”

method is called to confirm the update. The recordset and database connection are then closed.

#### *J. Image Manipulation*

Now that the database has been updated the original image must be copied, renamed, and reformatted to four different locations. The first action is a simple copy and rename accomplished by using the method “CopyFile” from the “FileSystemObject” instance “fs”. The syntax for this command is: fs.CopyFile originalfile, destinationfile. In this case, the code copies the current file being processed (x.name) to the “Actual” folder and gives it the file name found in the variable “lastID” along with the extension “.tiff”.

#### *K. Medium, Low, and Thumbnail Conversion*

The next section of code uses the software program ImageMagick® to convert the file from the TIFF format to the JPEG format. This is done by utilizing the command line functions of the windows system to call the application and its parameters. The first step is to define the variables needed to open the windows shell to access the command line prompt.

The next two steps assign the location of the executable convert.exe (part of ImageMagick®) and the location of the file to be converted to the variables “imagemagick” and “orgTiff”. The file to be converted in is the file copied in the last step from the “Actual” folder.

An instance of “WScript.Shell” is created and assigned to “WshShell”. This object provides functions to read windows system information, variables, work with the registry and in this case, access the command line prompt.

The next three sections of code perform the same routines while utilizing different parameters. The first section is the “medium” routine which copies the image file from the “actual” folder, converts it to the JPEG format, and places the new image into the “Medium” folder. The location of the “Medium” folder and the desired new file name are assigned to the “desJog” variable. Then the variables assigned to “imageMagick”, “orgTiff”, and “desJpg” are joined and assigned to the variable “Command Line”. The next line of code uses the “Exec” (execute) method of the “WshShell” object to execute the arguments in the variable “CommandLine”.

The next two sections of code perform the same steps as used in the medium image conversion to convert and copy the low and thumbnail images with the added arguments of resizing to “50%” and “15%” added to the command line. The variable “desJpg” is also set to the “Low” and “Thumbs” folders as destinations for the converted images.

Once these conversions have been completed, the memory for the objects called is freed by setting the instances to “Nothing”.

#### *L. Image Loop Completion*

Once the conversion of the images is complete, the current image file being processed (x.name) is deleted. This is accomplished using the “Delete” method of the “FileSystemObject” instance “fs”. This completes the image processing loop. The code now returns to the start of the loop to determine if there are any remaining images in the “original” folder to process.

### Final Steps

At the completion of the image processing, the memory held by “fo”, “fs”, and “fdb” are freed up by setting them to “Nothing”. The date and time of the update is then written to the browser screen utilizing the `response.write()` command. The original HTML comment tag is then closed (`-->`) and the ASP tag is closed (`%>`).

## III. RESULTS

### A. Trial Code

The trial implementation was performed utilizing a copy of the “Archives” folder renamed to “mmt”. This enabled the team to build the code a function at a time and track the output via the VBScript function “`response.write`”. This function allowed the developers to track variables and interactions with the database to follow the changes in data and image positions within the folders. Once the code was proven to be fully operational, a total of five variables were updated and the tracking code stripped from the final version. This ASP file was then placed into the “Pictures” folder in the main “archives” folder for full execution.

### B. Folder Permissions

It was found during the implementation of the final code that the folder permissions on the server needed to be changed to allow read/write permission. This setting allows ImageMagick® to place converted images in their proper locations.

### C. Update Notification

To allow for quick verification of an update attempt, a new table within the TeraScan database was created. The ASP script was modified to record the update time to this table and the front page of the CERSER website was set to display the latest attempt to update the database. This notification insures that the script is still being called daily to process images.

## IV. FUTURE WORK

### A. Modify Records

The first step that should next be taken is to create a page that would allow for modification of current records. The development should take place utilizing password security and allow the individual to modify, delete, or add records to the database as needed. Comments should be heavily utilized to ensure continuity once the current developer is replaced. The development of this page would also allow the addition of descriptions, events, and product information which were beyond the scope of this project.

### B. Email Notification

It is recommended that an email notification system be implemented to notify individuals that the database has been updated and new images are available. This would allow researchers and others to access new images immediately once they have been uploaded to the CERSER server. The format could be a “sign-up” form with the option of canceling

the notifications at any time. A notification system would also allow CERSER to notify individuals of upcoming events such as lectures or training sessions.

## REFERENCES

- [1] W3Schools ASP Tutorial, Retrieved December 5, 2005 from the World Wide Web: <http://www.w3schools.com/asp/default.asp>
- [2] W3Schools ADO Tutorial, Retrieved January 15, 2006 from the World Wide Web: [http://www.w3schools.com/ado/ado\\_intro.asp](http://www.w3schools.com/ado/ado_intro.asp)
- [3] IBM - Graphics from the Command Line, Retrieved January 15, 2004 from the World Wide Web: <http://www-128.ibm.com/developerworks/library/l-graf/?ca=dnt-428>
- [4] ImageMagick®, Retrieved January 8, 2006 from the World Wide Web: <http://www.imagemagick.org/script/index.php>
- [5] Dynamic-CD: Database Scripts, Retrieved January 21, 2006 from the World Wide Web: <http://www.phdcc.com/dynamic-cd/dcdoc/dbscript.asp>
- [6] “Active Server Pages 3.0”, maranGraphics, Inc. (2000)
- [7] Tutorial-Web - FileSystemObject, Retrieved February 16, 2006 from the World Wide Web: <http://www.tutorial-web.com/asp/fso/index.asp>
- [8] Active server pages tutorial: What are Active Server Pages?, Retrieved November 16, 2005 from the World Wide Web: <http://www.asptutorial.info/learn/Introduction.asp>
- [9] Active Server Pages and ASP PowerASP is all about Active Server Pages, Retrieved November 16, 2005 from the World Wide Web: <http://www.powerasp.com/>