**Paper CC-23**

# The Short-Order Batch

Carol A. Martell, UNC Highway Safety Research Center

## ABSTRACT

A project that documents pedestrian and bicycle crash attributes and locations requires manual data entry. The image of each crash report form, contained in a TIFF file, must be examined to categorize the crash attributes, and to geo-locate the crash. We equip the person performing this task with a Google Earth KML file showing the approximate location, and an Excel worksheet with other pertinent information. Each year, there are approximately 3500 of these crashes, requiring an unknown number of coders. In the past, a single, large KML file and Excel worksheet were created for the coder. When more coders were hired, the existing files had to be split and a fortune-teller predicted what amount of data to allocate to each coder. Further divisions were always required, and re-assembly was ugly.

The SAS® code we examine makes little molehills out of a mountain of raw data. When the coder is ready, a small batch of information appears. The code discovers filenames from the operating system, creates a batch folder, places an Excel worksheet and a SAS program in the batch folder. The user then customizes and runs the new SAS program to create the KML file.

## INTRODUCTION

This “batch-on-demand” SAS code allows for an orderly division of work. Unfortunately, there is no way to avoid the manual review; it cannot be accomplished algorithmically. We use SAS to manage file locations, to control naming conventions,to prepare the working files and to extract the results. The solution minimizes pain for both the coder and the analyst.

## THE DATA

A list of some 3,500 candidate IDs is identified. This list is used three times.

First, an Oracle database is queried to pull information from various tables for the ID list. The query results are placed into an Excel spreadsheet, one record per line.

Second, the list of IDs is used in a batch request for image files. The image files contain one or more pages of information about each ID. The provider packages the images into a numbered series of multipage TIFF files. The IDs are in numeric order within the TIFFs, but the number of IDs per TIFF varies. Consequently, the ID range contained within a single TIFF is not known until the file is opened.

Third, the ID list is used to query Oracle and create an approximate street address for geocoding, storing the results in a SAS table. The geocoded addresses are later used to create the approximate placemarks in a KML file. These three items, the Excel spreadsheet, the TIFF images, and the KML file are used for coding

## WHY CHANGE?

In the past, a single coder had worked with the files. There was one Excel worksheet and one KML file. Beginning with the first TIFF file, the image IDs aligned one-for-one with those in the Excel and KML files. When the coder was finished, the analyst could easily extract the results from the finished files.

Mid-project, the workload was increased, a new coder was hired and trained, and the working files were split between them. The best-guess point for the split was about two-thirds/one-third, taking the coding rates and remaining work into account. After some time, a third coder had to be hired, and the two-thirds/one-third files were further divided. More training for the new coder slowed the first coder down, creating the need for even more file divisions. In other words, chaos ensued, and extracting results was difficult and time-consuming.

In order for the final extraction process to be flexible yet algorithmic, a new approach was needed.

## THE SOLUTION

The solution is to avoid/control the chaos by designing a system that remains orderly regardless of the number of coders. We divide the data into batches before processing begins, creating a set of files for each batch. Normally, one would pick a constant number of cases – 25, 40, 50 – for the batch size. In this situation, we already have the data divided into smaller chunks in the TIFF files. Unfortunately, the ID range within a TIFF is unknown, and the

batch-sized KML and Excel worksheets cannot be prepared without knowing the ID range. We could have someone open every TIFF file and record the ID range and filename for each of the TIFF files, then use that information to prepare the batch files. Unfortunately, in a project already rife with tedium, this is not an enhancement. Fortunately, it is not painful to open one TIFF file every now and then! This act of kindness led to creating a system wherein SAS creates a batch working directory and prepares a set of working files.

## THE CODE

Two SAS programs are needed – one to identify the next TIFF file, create a directory, move the TIFF into that directory, and write a second SAS program in the directory. That SAS program includes code that prepares the Excel and KML files for the batch. Before executing that second program, the coder opens the TIFF to find the ID range for the batch. Next, the coder edits that second SAS program, providing the ID range.

Here is the code for the first SAS program. This is executed by a coder in need of a new batch of files. We will next examine this more closely.

```
options obs=1 noxwait;
data next;
filename x PIPE 'dir /b T:\MAINDIRECTORY\tifs\*.tif';
infile x;
input;
tifname=scan(_infile_,1,' ');
prefx=scan(tifname,1,'.');
call symput('tifnam',tifname);
call symput('prefx',prefx);
run;
data _null_;
x "mkdir T:\MAINDIRECTORY\batches\%trim(&prefx)";
x "move T:\MAINDIRECTORY\tifs\&tifnam T:\MAINDIRECTORY\batches\%trim(&prefx)";
file "T:\MAINDIRECTORY\batches\%trim(&prefx)\%trim(&prefx).sas";
put %nrstr("%let beg=;");
put %nrstr("%let end=;");
put %nrstr("%let tif=") "%str(%trim(&prefx));";
put 'options mautosource sasautos="T:\MAINDIRECTORY";';
put %nrstr("%setup;");
run;
```

We examine the above code. The OBS are set to 1 because we only need one TIFF filename. Because we will be shelling out to the operating system the XWAIT option would cause command windows to remain open until we manually closed them. We change that option to NOXWAIT.

```
options obs=1 noxwait;
```

In our DATA step, we PIPE the results of an operating system command to SAS as our infile. Since we have set OBS to 1, we will retrieve one TIFF filename. Because we use the /b switch on the dir command, no information other than the filename will appear in the INFILE.

```
data next;
filename x PIPE 'dir /b T:\MAINDIRECTORY\tifs\*.tif';
infile x;
```

Execution of the INPUT statement places the TIFF filename into the automatic variable _INFILE_. We use the SCAN function to get rid of any trailing blanks, and assign the result to the variable TIFNAME. We strip the extension from that variable and assign the result to the variable PREFX; we will use that as the batch directory name. We place each value into macro variables, &TIFNAM and &PREFX.

```
input;
tifname=scan(_infile_,1,' ');
prefx=scan(tifname,1,'.');
call symput('tifnam',tifname);
call symput('prefx',prefx);
run;
```

Our next DATA step begins by creating the new batch directory and moving the TIFF file to that directory

```
data _null_;
x "mkdir T:\MAINDIRECTORY\batches\%trim(&prefx)";
x "move T:\MAINDIRECTORY\tifs\&tifnam T:\MAINDIRECTORY\batches\%trim(&prefx)";
```

The second task of the DATA step is to write a SAS program in the directory.

```
file "T:\MAINDIRECTORY\batches\%trim(&prefx)\%trim(&prefx).sas";
```

This program begins with %LET statements for the begin and end values of the ID range.

```
put %nrstr("%let beg=;");
put %nrstr("%let end=;");
```

Another %LET supplies the TIFF file basename to the program.

```
put %nrstr("%let tif=") "%str(%trim(&prefx));";
```

A SAS macro that will prepare the Excel and KML files is located with the SASAUTOS option, and the macro is invoked.

```
put 'options mautosource sasautos="T:\MAINDIRECTORY";';
put %nrstr("%setup;");
run;
```

## CONCLUSION

Preparing for the unknown is always good practice, as is minimizing pain for your users. You might find that by minimizing work for others you also minimize work for yourself. While this project mostly involves the use of other software packages, SAS remains at the core to facilitate the workload.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Carol Martell
UNC Highway Safety Research Center
Chapel Hill, NC
carol_martell@unc.edu