Paper 094-2008

Concatenating Inputs Using SAS® on the PC

Jim Noe, TEK Systems Inc, Leawood, KS

ABSTRACT

Sometimes it is necessary to read a large number of flat files with as little coding as possible. If the files all contain the same kind of information in the same format then the job can be greatly simplified by placing the files in a common folder with names that are similar. We refer to this folder in Base SAS® using the FILENAME statement to define the path to the folder. Then in the DATASTEP we use the INFILE statement with a wildcard character such as an asterisk. This paper will show how to do this with the minimum amount of effort.

INTRODUCTION

Suppose we are working for a global travel agency with a website that they use to do a satisfaction survey periodically. The data collected on the Web form is transmitted to the company in the form of an e-mail when the person being surveyed clicks on the submit button. Each e-mail message then becomes a separate response data file. For each survey there are a variable number of these response data files. There can be up to 500. The example for the purposes of this paper has only five response files with different file names.

The files all have variable-length records. Each record contains only text. The lines in the files contain similar data in all files and the formatting is the same. Line 1 contains the city where the subject lives. Line 2 contains the estimated miles each subject drives per week to commute to work. Line 3 contains the subject's choice of a vacation destination. Line 4 contains the amount of vacation the person can take per year. Line 5 contains the answer to whether the subject would prefer to fly to their destination when they take vacation. Line 6 contains the response to a question about the importance of saving time when scheduling travel. Line 7 is the response to a question regarding how often vacations need to be rescheduled in their workplace. Line 8 contains the subject's estimated number of times they are involved in vacation scheduling changes per month. It is possible for one or more of the responses to be blank which should result in a missing value set for that line on the output.

So, the only identifying characteristic for each response is the line number it is on. But all files have the same responses on the same line numbers. We will have to capture the line number while reading the files. We also need to capture the name of the file the responses are coming from because later on we may want to capture the date and time the file was received using the file name.

We want to be able to read the data from all of these files all at once, without having to specify each file's fully qualified path and filename. Is there an easy way we can to this in Base SAS®? Yes. We can concatenate the input files using SAS. Let's see how it is done.

METHOD

First we will put all of the data files into a single folder, by study-id, segregated from all other study files with filenames that might be misinterpreted to be part of the same study. The name assigned to the folder for the example survey is 'CT$12 Survey'. Other surveys may have similar folder names but for each survey the folder name is unique.

The FILENAME statement now points to the folder 'CT$12 Survey' and looks like this:

```
filename concat 'c:/documents and settings/jim noe/ct$12 survey';
```

Now we can create the DATA step that will read the concatenated input files. There will be two numeric variables, one for counting the line number, and the other to determine the length of the input line. There will be three character variables with a maximum length of 200 bytes. One to save the file name of the current file, one to save the text of the line read and a temporary one to use for the input file name variable on the INFILE statement. Notice that the infile statement uses folder CONCAT with the member name "*.txt" wildcard. This causes all files ending with ".txt" to be read from the folder specified in the filename statement above. We also have to retain the Input_File_Name in order to detect when the name changes. Then we retain the line_nbr so we can use it for counting the number of lines read.

HERE IS THE DATA STEP:

```
/* SAS code begins here */
 data contents;
     length line_nbr var_len 8;
     length line in_file Input_File_name$ 200;
     retain Input_File_name ' ';
     retain line_nbr 0;
/* The following statement sets the maximum length of the line to be read when the infile statement is executed.*/
     var_len = 200;
     infile concat(*.txt) filename=in_file   missover length=var_len;
/* Note the missover option which will keep SAS from going to the next line for data if the current line is blank.*/
     input @ 01 line $varying200. var_len;
/* Check if the file name has changed so we can reset the line counter and save the new filename. */
     if Input_File_name NE in_file
     then do;
       line_nbr = 0;
       Input_File_name = trim(left(in_file));
       end;
    line_nbr + 1;
    line = trim(line);
    line_len = var_len;
 run;
/* Add a PROC PRINT to see what the resulting data looks like. */
 proc print data=contents(obs=40);
 run;
/* SAS code ends here */
```

CONCLUSION

To implement this solution to the problem of reading multiple input files on the PC, remember there are two steps. The first step is to place all of the input files in a single folder with similar but unique file names. The second is to use the FILENAME statement in a Base SAS® Datastep to refer to the folder and use the INFILE statement with a wildcard asterisk, (*), to refer to the parts of the input file names that are not common.

CONTACT INFORMATION

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

Jim Noe
TEK Systems, Inc.
12009 Ash Street, Apt 27
Leawood, KS 66209-3106
Work Phone: 913-339-6105
E-mail: jamesknoe@earthlink.net