Paper 102-2011

Emailing from Your SAS® Program to Keep Everyone Informed

Bob Bolen, Southern Company, Atlanta, GA
Diane Cunningham, Southern Company, Atlanta, GA

ABSTRACT

Have you ever had a long running SAS® job that ends early with an error? Wish you would have known about it? Would you like to have the program send the results of a run to your client? We will look at a technique for emailing from your existing SAS code to do just that.

INTRODUCTION

Southern Company is an electric utility holding company. Its operating utilities are Georgia Power, Alabama Power, Gulf Power and Mississippi Power. These operating companies serve over 4.3 million customers in a four state area. The accounting master files for the companies contain well in excess of 100 million rows of monthly billing information that is regularly processed by the Load research & Customer Analytics Department on one of our UNIX systems. Many of the programs run by the department take several hours to complete. Sometimes, when a job step would abort, the analysts might not know it right away because they would check the job when they expected it to finish. This was a waste of time and resources. Learning how to have our programs be more efficient in talking to us when a problem exists or even sending distributions of information on job completion has saved us many hours of lost time. The tips in this paper on using your email server in conjunction with SAS should help you in your work as well.

YOUR E-MAIL SERVER

The first thing you should check is if the options **EMAILSYS** and **EMAILHOST** have been set up in your configuration file. If you are still running SAS Version 8, that file is sasv8.cfg, in version 9 that file is sasv9.cfg located where SAS was installed. In your UNIX home directory, you could also have a local copy of the configuration file, such as sasv9_local.cfg like I do. These options could also be run at invocation of SAS as well, but are probably better located in the configuration file. The syntax of the two statements is shown below.

```
/** email options  **/
-emailsys smtp
-emailhost <your email server>
```

Another option that might have to be specified is **EMAILPORT**, which gives the port number used by your email host server. More information on this option can be found in the SAS Help and Documentation by searching for **EMAILPORT**. My UNIX system did not require this option, but check with your UNIX administrator for additional guidance.

E-MAIL PROGRAM ERROR

The following code example shows the code required to e-mail a message to you if your program has an error in one of its data sets.

```
(main body of code to create data set readings)

filename dbfmail1 email "bebolen@southernco.com"
    subject="register_read6.sas errors" ;
```

The FILEREF dbfmail1 specifies who the email will be sent to, in this case, my address bebolen@southernco.com. It also creates the subject line, with the SUBJECT= option, for the email telling us that an error occurred in the register_read6 program. Further options available for EMAIL engine of FILENAME statements and complete syntax can be found in the SAS Help and Documentation by searching for filename and email.

Once the FILENAME statement has been processed by SAS, the program moves on to coding designed to test for certain conditions in the variables in the data set. In the DATA _null_ step below, we can see one such example. The readings data set is brought in and the variable difdatechk is tested for a value less than 0. If the program finds this condition to be true, it passes into the then do. A FILE statement points back to the FILENAME statement processed earlier, then PUT statements are used to build the body of the email that will be sent back to the analyst running the code.

```
Data _null_ ;
  Set readings ;
 if difdatechk lt 0 then do ;
   file dbfmail1 ;
   put 'The program has detected an error in the data set readings'  ;
   put 'The value of difdatechk is less than 0 for observation ' _n_ ;
 end ;
run ;
```

The body of this email tells us an error was found in the data set readings on observation _n_ where the variable difdatechk is less than 0. Below shows the e-mail that was sent from a run of the program on UNIX.

```
To: Bolen, Bob E.
Subject: register_read6.sas errors

The program has detected an error in the data set readings
The value of difdatechk is less than 0 for observation 1
```

E-MAIL PROGRAM OUTPUT

Another use of using an email output from a SAS program is to take a completed data set, turn it into a transportable file and email it as an attachment to an end user directly from a running program on UNIX. In this example, we have processed code and created the data set **total2**. The client wants use to send them a file that they can use in Excel. For this program, a DBF file was chosen. The PROC DBF outputs a db4 file to the FILENAME **outfile** specifies a valid UNIX location and file name with the DBF extension.

```
   (main body of code to process data set total2 for output to dbf file)

filename outfile '/temp/test_for_jan98_2008.dbf';
proc dbf db4=outfile data = total2;
```

Then the FILENAME dbfmail 1 is used to create the email TO:, jrjohann@southernco.com and CC:, in this case, my email address. The subject line of the email is created in the SUBJECT= option while the attachments are specified in the ATTACH= option. IN this example, there are two attachments, first the actual DBF file and then the log file created by this program. Once again, the data _null_ is used to create the text of the email and point to the file dbfmail1.

```
filename dbfmail1 email "jrjohann@southernco.com"
    cc=("bebolen@southernco.com")
    subject="DBF files from program 'jan98_2008.sas;"
```

```
    attach=("/temp/jan98_2008.dbf" ct="application/octet-stream"
            "/temp/jan98_2008.log") ;
data _null_ ;
 file dbfmail1 ;
 put 'Here is the monthly DBF file for the program   jan98_2008.sas'  ;
 put 'Please let me know if there are any problems with the outputs' ;
 put ' '   ;
 put 'Bob Bolen'  ;
 put '8-506-3730'  ;
run ;
```

CONCLUSION

The use of email in the UNIX/SAS environment has allowed us to start jobs and not worry about having to come back and check them errors or spend additional time transferring files created in the code after the job has completed. Our work proceeds faster and out clients get their outputs faster.

AUTHOR BIO

Bob Bolen has worked in the electric utility industry with Southern Company for over 30 years. In his current position as Technical Project Manager in the Load Research department, he is responsible for statistical and data analysis perform by his team. Bob has been using SAS for over 27 years and was conference co-chair for SESUG 2009, serves on the Executive Council of SESUG and has spoken at various levels of user groups.

CONTACT INFORMATION

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

Bob Bolen
Southern Company
Bin 10206
241 Ralph McGill Blvd, NE
Atlanta, GA 30308
Work Phone: 404-506-3730
E-mail: bebolen@southernco.com