

Paper CC-155

Talk to Me!

Elizabeth Axelrod, Abt Associates Inc.

ABSTRACT

Wouldn't it be nice if your long-running program could tap you on the shoulder and say 'Okay, I'm all done now'. It can! This quick tip will show you how easy it is to have your SAS® program send you (or anyone else) an email during program execution. Once you've got the simple basics down, you'll come up with all sorts of uses for this great feature, and you'll wonder how you ever lived without it.

INTRODUCTION

Sending an email from within a SAS program is surprisingly easy. A few lines of code are all you need. Once you get the basics down you can enhance your code to add flexibility and to make your emails more informative.

THE BASICS

The syntax is quite straightforward: start with a FILENAME statement using the EMAIL engine, and a recipient's email address:

```
FILENAME fileref EMAIL "recipient email address goes here";
```

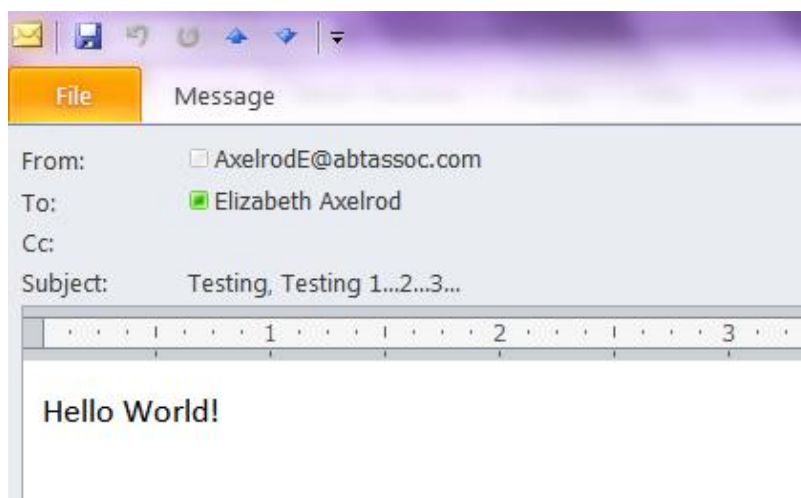
Then use FILE and PUT statements in a DATA step:

```
DATA _null_;  
  FILE filreref;  
  PUT "your message goes here";  
RUN;
```

The PUT statement provides text for the body of your email. In the following example, I'm sending an email to myself. Note that I've added a subject line.

```
FILENAME myemail EMAIL "elizabeth_axelrod@abtassoc.com"  
                  SUBJECT='Testing, Testing, 1...2...3';  
  
DATA _null_;  
  FILE myemail;  
  PUT "Hello World!";  
RUN;
```

When I execute this code, I receive this email in my inbox:



Display 1. Email sent from within SAS

Here is the SAS log:

```
NOTE: The file MYMAIL is:
      E-Mail Access Device

Message sent
  To:      "elizabeth_axelrod@abtassoc.com"
  Cc:
  Bcc:
  Subject: Testing, Testing 1...2...3...
  Attachments:

NOTE: 2 records were written to the file MYMAIL.
      The minimum record length was 23.
      The maximum record length was 24.

NOTE: DATA statement used (Total process time):
      real time          5.23 seconds
      cpu time           0.00 seconds
```

Output 1. LOG from code using the EMAIL fileref.

Just by looking at the log for this code, you get a hint of other things you can do with this. By specifying various options on the FILENAME statement, you can send emails to multiple recipients, copy recipients, and send attachments.

```
FILENAME fileref EMAIL "recipient email address goes here"
  TO =      "add additional recipients here"
  CC =      "add more recipients here"
  SUBJECT = "my subject line here"
  ATTACH = "pathname/filename";
```

And of course, you can tailor this in many ways to make it more useful, like using macro variables in the FILENAME or the DATA step to make your code more flexible.

WILL THIS WORK FOR YOU?

Yes! (Probably.) This works on any platform with base SAS, and no special add-ons are needed. When I first tried this, I was convinced it wouldn't work, and that I would need to do something special first. But I've used this simple technique successfully on various platforms including a Windows SAS server, a mainframe, and a Unix server.

If you look at some of the online papers on this topic, you'll find that they sound pretty technical, and go into details that are appropriate mainly for SAS system administrators. So unless *you* are the system administrator (i.e., you're using an individual installation of SAS), it is most likely the case that your system administrator has provided the network name of the email server, so the EMAIL engine is already defined and specified, and you won't need to worry about it.

MAKE IT BETTER

I'm going to show you how I've tailored this simple method for my use. I like to send myself an email after my long-running job is complete, telling me which program just ended, and if it ended successfully. First I need to get the program name, but I don't want to hard-code it. This handy function will resolve to the full pathname and program name you're running.

```
%let pgm = %sysfunc(getoption(sysin));
```

It's important to note that this only works if you're running your job in BATCH mode. You won't get an error if you run this interactively, but it won't resolve as you'd like it to. If you just want to grab the program name – without the pathname - you can then use the %scan function below, making sure to specify appropriate delimiters for your system:

```
%let pgm = %scan(&pgm,-2,'.\');
```

Here's the FILENAME statement, where EOJ just stands for End of Job, and &PGM will resolve to my program name:

```
FILENAME mymail EMAIL 'elizabeth_axelrod@abtassoc.com'
  subject="EOJ: pgm=&PGM";
```

Now I need to figure out if my program ran successfully. The automatic macro variable &SYSCC will resolve to 0 if your program ran with no errors or warnings, or some non-zero number otherwise. The code below will send an email with one message if it ran successfully, and a different message if it failed.

```
data _null_;
  file mymail;
  if &syscc=0 then
    put "Program &pgm ended successfully";
  else do;
    put "Program &pgm ended in pathetic failure";
    put "syscc=&syscc";
  end;
run;
```

Using this code at the end of my program, I'm hoping to receive an email which tells me that my job completed, and whether it ran successfully or failed. The problem is that this code only works if my program ran successfully; if there was an error earlier the program, I never receive an email message. What's going on here?

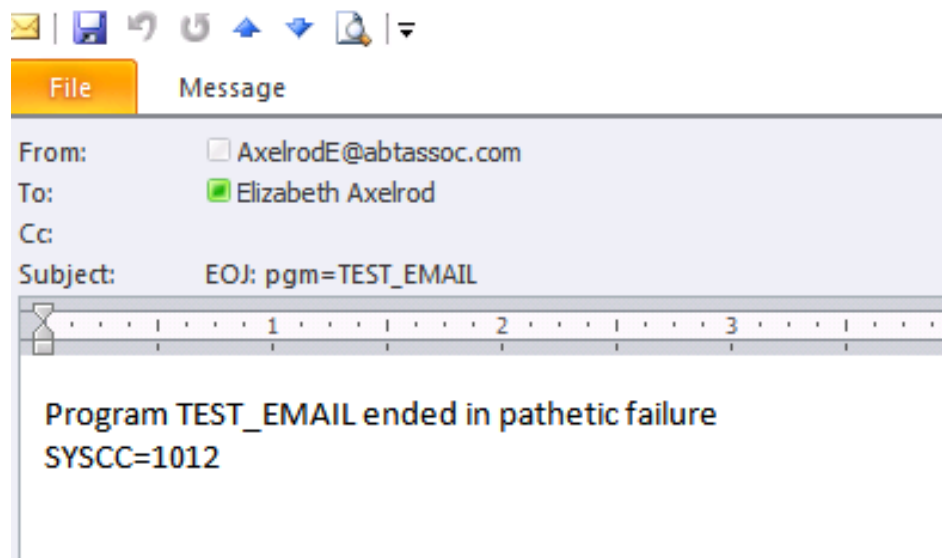
When SAS encounters an error, the default behavior is to set OBS=0, stop executing code, but to continue checking for syntax errors. This behavior is controlled by the system option SYNTAXCHECK, which is set as the default. And this is generally a good thing – we don't want SAS running ahead, executing steps willy-nilly, if SAS has already encountered an error. But in *this* case, I *do* want SAS to run that DATA step. To get this DATA step to execute - even if SAS encountered an earlier error - I preface the DATA step above with this OPTIONS statement:

```
options nosyntaxcheck;
```

This tells SAS to go ahead and execute the DATA step that follows, even if previous errors were encountered.

It's important to note that I'm only telling SAS to go into NOSYNTAXCHECK mode at the very end of my batch program. Also, the sole purpose of the final DATA step is simply to send an email - not to process data. I mention this because the behavior of the SYNTAXCHECK/NOSYNTAXCHECK option is more nuanced than presented here, and I encourage you to carefully review "*An Overview of Syntax Check Mode and Why it is Important*", referenced at the end of this paper before tampering with this option - especially if you want to reset it during your program. For my use, since I'm putting this code as my last step in a program which I run in batch mode, I feel that it's a safe method to use.

Now when I run this code from a program with errors, here is the email I find in my inbox:



Display 2. Email with job success or failure status revealed.

For curiosity sake, I included the value of the &SYSCC macro variable in the email message. You can look up the different values for &SYSCC, but no matter what, you'll want to review your SAS Log. This is of course true even if your job ran with no errors... because we all know that just because your program ran error-free, there's no guarantee that your results are actually what you wanted.

CONCLUSION

Sending an email from within a SAS program is a valuable tool that's easy to code. Tailor it to make it your own!

REFERENCES

- Worden, Jeanina. Jones, Philip. *"You've Got Mail – E-mailing Messages and Output Using SAS® EMAIL Engine"*. 2004. Available at <http://www2.sas.com/proceedings/sugi29/178-29.pdf>.
- Tilanus, Erik W. *"Sending E-mail from the DATA step, by Erik W. Tilanus"*. 2008. Available at <http://www2.sas.com/proceedings/forum2008/038-2008.pdf>
- Billings, Thomas E. *"An Overview of Syntax Check Mode and Why it is Important"*. Available at <http://support.sas.com/resources/papers/proceedings13/327-2013.pdf>.
- support.sas.com

ACKNOWLEDGEMENTS

The author wishes to thank Paul Grant for suggesting the nosyntaxcheck option, and for reviewing and editing this paper.

CONTACT INFORMATION

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

Elizabeth Axelrod
E-mail: elizabeth_axelrod@abtassoc.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.