

Paper 178-29

You've Got Mail – E-mailing Messages and Output Using SAS® EMAIL Engine

Jeanina Worden, Mid America Heart Institute, Kansas City, MO

Philip Jones, Mid America Heart Institute, Kansas City, MO

ABSTRACT

This paper will cover the syntax of the FILENAME and FILE statements to automatically send custom e-mails and files, based on the results of a program, using the FILENAME EMAIL access method. SAS users of any level who have information to send electronically can use the information in this paper. Though the syntax used in this paper applies to SAS Version 8.2, these features are available with SAS Release 6.11 and beyond.

WHAT IT'S ALL ABOUT

So you have a data coordinator that, on a weekly basis, would like to review an updated version of a report you programmed. Maybe there is someone who would like to be notified if a batch process had issues. How much time is spent manually notifying the client that data is or is not ready? Wouldn't it be nice if a message or the report would automatically be e-mailed to the person with the status when the program finished running? With the SAS E-mail engine, you can perform all of these tasks and more. By using the FILENAME EMAIL access method and a conditional DATA step, you can automate the sending of messages and files based on the results of your program. This paper will cover the syntax of the e-mail options used within the FILENAME and FILE statements to address the recipient(s) of the e-mail and the attachments to be included. It will also cover the use of PUT statements to build what the text message will say. Then it will describe the use of conditional statements to determine when to send specific information to different recipients. Though the syntax used in this paper applies to SAS Version 8.2, these features are available with SAS Release 6.11 and beyond.

START YOUR ENGINES

While there is no physical key to start this engine, there are key statements that need to be in place to start using the EMAIL access method. Before sending electronic mail either interactively or programmatically using SAS statements, the engine must be initialized. In short, this means to use the EMAIL engine the configuration file may need to be customized for your e-mail application. Some of these options are used to specify the e-mail interface (-EMAILSYS), the native e-mail interactive dialog interface (-EMAILDLG), e-mail login or profile of the underlying e-mail system (-EMAILID), and the password (-EMAILPW) to be used. This paper assumes these customizations are already in place. However, to learn more, online documentation is available that will give the authorized users all the information needed to set up the system for e-mailing.

Once the e-mail feature is set up, the SAS system will recognize the customization as the default system options. For instance if the EMAILSYS = MAPI, the default would be for Microsoft Mail and Microsoft Exchange. That said, any option put into the FILENAME statement would override these defaults. The generic syntax for the FILENAME statement is:

```
FILENAME fileref EMAIL 'address' <e-mail-options>;
```

To include the modifications to the EMAIL options, the FILENAME statement would look something like this:

```
FILENAME fileref EMAIL 'address' <e-mail-options> EMAILSYS=<e-mail interface to use>
EMAILPW=<password of user> EMAILID=<userid>;
```

The code this paper is based on uses the FILENAME statement to make modifications to the default profile name, EMAILID, is:

```
FILENAME sugi EMAIL 'jworden@saint-lukes.org' EMAILID='Microsoft Outlook';
```

Whether the system default options are used or modified within the FILENAME, it is important to note a few caveats before continuing. First, if the name or password contains spaces, you must use double quotes. Next, if the EMAILID and the EMAILPW are not specified, and a user is not logged into the e-mail system, SAS will prompt for a user name and password. In the case of batch programs, this will cause a program to stall without warning. Finally, the user who is logged onto the system will by default be recognized as the sender of the e-mail. Specifying the EMAILID and EMAILPW is an easy way to override the default profile, however security issues need to be considered before coding someone's USERID and password into a program.

TO WHOM IT MAY CONCERN

Now that the engine is humming along let's get to the "who", "what" and "when" of the matter. First the "who"; let's look again at the FILENAME statement:

```
FILENAME sugi EMAIL 'jworden@saint-lukes.org' EMAILID='Microsoft Outlook';
```

The FILENAME associates a SAS *fileref*, named here as SUGI, with an external file or output device. So SUGI is temporarily assigned an aggregate storage location that the SAS system identifies. EMAIL is the device-type keyword to let SAS know you want to use e-mail. Then the address portion of the statement identifies the e-mail address of the person your sending the e-mail to. The FILENAME statement, along with a DATA step containing a FILE statement that references the *fileref*, is all you need to send e-mail:

```
FILENAME sugi EMAIL 'jworden@saint-lukes.org' EMAILID='Microsoft Outlook';
DATA _null_;
FILE sugi;
RUN;
```

However, if you run this code you would find there are some very important things missing. No subject is identified, which would be a problem if the recipient or his company automatically deletes messages without subjects. Also, there is nothing in the body of the e-mail, and what was that about sending the SAS data with the message? Where is it? That's where you've got options, e-mail options that is.

The e-mail options will look familiar to anyone who has used an e-mail application. The options include: TO= *to-address*, CC= *cc-address*, SUBJECT= *subject*, and ATTACH= *filename.ext*. These options are explained below in more detail and can be used within the FILENAME statement or the FILE statement. However, if used within the FILE statement, the option will override those specified in the FILENAME statement and, unless otherwise stated, must follow the same syntax rules. If the assignment contains more than one word or special any characters, you must enclose it in double quotes. If you want to make more than one assignment for an option you must enclose the group in parentheses and each individual assignment in double quotes.

TO=

The TO option specifies the primary recipient(s) of the e-mail. In the program above, the recipient of the e-mail was specified in the address portion of the FILENAME statement. Using the TO option within the FILE statement is a handy way of changing the recipient if using conditional statements to send the e-mail. The code would look like this:

```
TO= 'jworden@saint-lukes.org';
TO= ('jworden@saint-lukes.org' 'someone@else.com');
```

You can also use an ID from your contact list when using your own mail application:

```
TO= 'Jeanina Worden';
```

CC=

This option specifies the recipient(s) of copies of the e-mail. When the primary recipient is someone other than yourself, using this code is a way for you to receive a copy of the information for your record. Valid CC values are:

```
CC= 'jworden@saint-lukes.org';
CC= ('jworden@saint-lukes.org' 'someone@else.com');
```

SUBJECT=

SUBJECT specifies the text that will appear in the subject line of an e-mail. As with the TO option, using this within the FILE statement allows the subject to change based on the recipient. NOTE: to use a single word for the SUBJECT no quotation marks are needed:

```
SUBJECT= Data;
SUBJECT= "Data for Review";
```

ATTACH=

The last option discussed here is the ATTACH option. By specifying the exact path of the file(s), you can send a report or a copy of a dataset each time a program is run with new data. The ATTACH syntax is:

```
ATTACH= 'C:\SAS Resources\SUGI29\example_report.sas7bat';
ATTACH= ("C:\SAS Resources\SUGI29\example_report.sas7bat" "C:\SAS
Resources\SUGI29\sugi_email.txt");
```

'PUT' IT IN YOUR OWN WORDS

Now that SAS knows who you're sending the e-mail to, what do you want to say? PUT statements write to an external file that is specified in the most recent FILE statement, in this case it writes the statements to SUGI. The use of PUT statements defines the text in the body of the e-mail. The PUT statements can contain one long text string or several PUT statements can be used to add formatting to the text. With this, the text of the e-mail can be customized depending on the result of the program. When combining the e-mail options and the PUT statements together within a FILE statement, it looks like this:

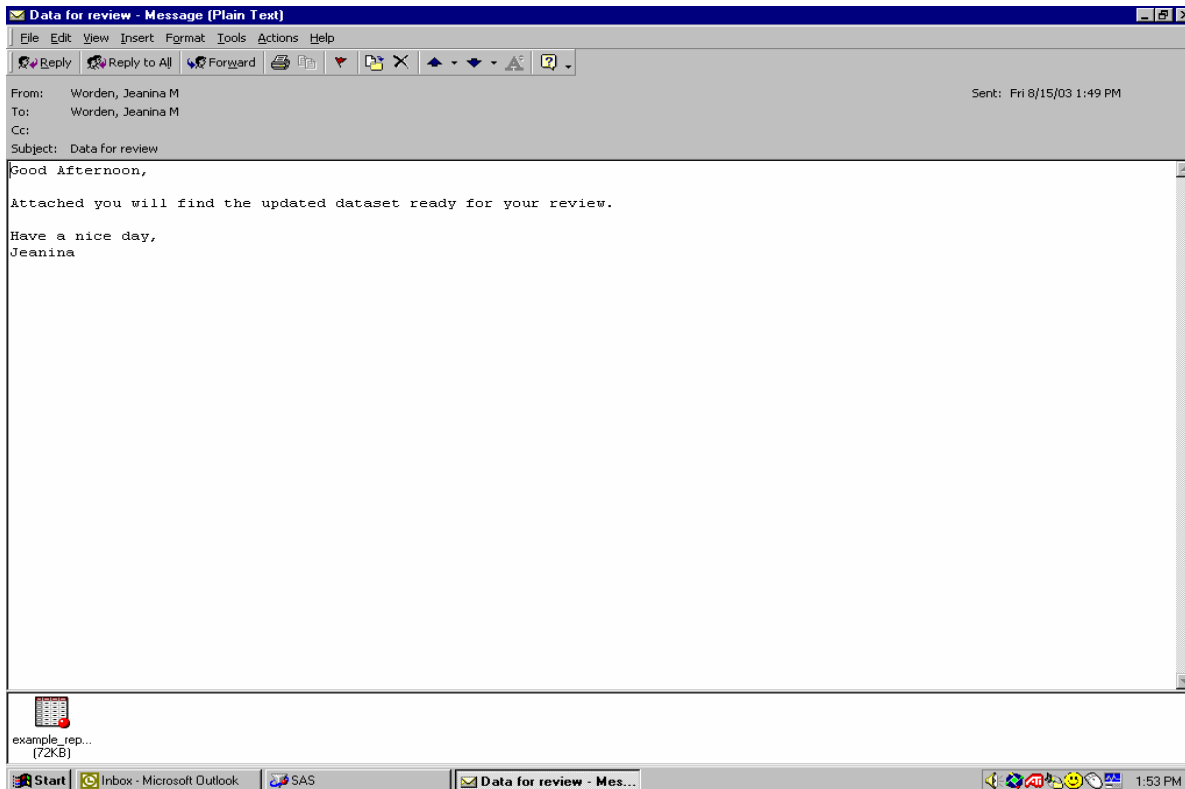
```
DATA _null_;
FILE sugi
SUBJECT= 'Data for review'
ATTACH= 'C:\WINNT\Profiles\A30885\Desktop\SUGI29\example_report.sas7bdat'
;
PUT 'Good Afternoon, ';
PUT ' ';
PUT 'Attached you will find the updated dataset ready for your review.';
```

```

PUT ` `;
PUT `Have a nice day,';
PUT `Jeanina';
RUN;

```

To give you an e-mail that looks like this:



Just a note, there are ways to modify the e-mail options within the PUT statements as well, however these options are not discussed in this paper. If interested, SAS customer support's documentation area at www.support.sas.com/documentation/index.html contains several white papers on them.

'IF' IT IS SOMEONE SPECIAL

The last thing to discuss is when to send the e-mail. The placement of the code within the program determines when the e-mail message is sent. It can be placed anywhere within the program after the dataset/report/log has been created. Well then how do you tell SAS if you want a status e-mail to be sent to one person or e-mail with an attachment to be sent to another person? By using conditional statements within the DATA step, you can specify options to customize what information is sent to each recipient based on which condition is met. The macro that this paper is based on uses the SYSERR macro variable to determine if the batch program completed successfully. The SYSERR is assigned 0 if there were no errors encountered but gets assigned 1 to ≥ 4 for errors ranging from warnings to an actual error. If the SYSERR=0 the dataset is sent to the address in the FILENAME statement with an attachment and message stating it's ready for review. However, since any error needed to be looked at, if the SYSERR $\neq 0$, a simple message e-mail is sent stating it needs to be looked into. Here's how it all came together:

```

libname sugi 'S:\';
FILENAME sugi EMAIL 'jworden@saint-lukes.org'
                emailid = "Microsoft Outlook";

%macro sendmail;
DATA testmail;
    set sugi.example_report;
run;
%if &syserr = 0 %then %do;
    data _null_;
        FILE sugi
        SUBJECT= 'Data for review'
        ATTACH= ('S:\example_report.sas7bdat'
                'S:\email_text.txt');

```

```

                PUT 'Good Afternoon,';
                PUT ' ';
                PUT 'Attached you will find the updated dataset ready for your
review. ';

                PUT ' ';
                PUT 'Have a nice day,';
                PUT 'Jeanina';

        run;
%end;
%else %do;
    data _null_;
        FILE sugi
        TO='jworden@kc.rr.com'
        SUBJECT= 'Error With Data';
        PUT 'ATTENTION PLEASE,';
        PUT 'There was an error when this program ran. Please check the
program and resubmit.';
        PUT ' ';
        PUT 'Jeanina';

    run;
%end;
%mend sendmail;
%sendmail;

```

THE FINAL WORD

The EMAIL engine is a powerful tool that can be used to directly contact recipients when a program has completed, with or without errors, eliminating a step the programmer must complete manually. Once the engine is initialized, through the configuration file or modified from within the FILENAME statement, the e-mail options allow for customization of who the intended recipient is to be, persons to be copied, the subject of the e-mail, add attachments, and specify text within the body of the e-mail. There are even options for the options; they can be used within the FILENAME statement or specified/modified within the FILE statement. Following simple syntax rules, and conditional statements, this code can be placed within any program to send e-mails to multiple recipients.

REFERENCES

SAS Institute Inc. (1996), SAS Companion for the Microsoft Windows Environment, Cary, NC: SAS Institute Inc.
SAS Institute Inc. (1995), SAS Software: Changes and Enhancements, Release 6.11, Cary NC: SAS Institute Inc.
SAS Institute Inc. (1997), SAS Macro Language: Reference, First Edition, Cary, NC: SAS Institute Inc.

CONTACT INFORMATION

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

Author Name: Jeanina Worden
Company: Mid America Heart Institute
Address: Saint Luke's Hospital, HI5
4401 Wornall Road
City state ZIP: Kansas City, MO
Work Phone: 816-932-3436
Fax: 816-932-5613
Email: jworden@saint-lukes.org

Author Name: Philip Jones
Company: Mid America Heart Institute
Address: Saint Luke's Hospital, HI5
4401 Wornall Road
City state ZIP: Kansas City, MO
Work Phone: 816-932-3273
Fax: 816-932-5613
Email: pgjones@saint-lukes.org

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.