

Stop waiting – Get notification email at the end of SAS® code execution using SAS® EMAIL Engine

Nirav Darji, GCE Solutions Inc.

ABSTRACT

DATA step and statements like FILENAME, FILE and PUT can be used for automatically sending custom e-mails. This utility can help programmers in cases where SAS codes take lot of time to complete the execution and we kept waiting for completion. It can also help when the teams are working in different time zones/shifts. The team working in another time zone or next shift can get notified once the execution gets completed. We will see how to send email with or without attachment (HTML, EXCEL, SAS files etc.) or personalized email and how to send email when it fulfills certain conditions.

INTRODUCTION

This paper describes that how you can use SAS EMAIL Engine to send email using DATA step. To send email using DATA step, you first have to define the SAS environment. Once the SAS environment is set, you can use SAS statements to send email. You can also send emails conditionally and attach different type of files in email.

INITIALIZING E-MAIL

SAS supports three types of electronic mail interfaces:

- MAPI (Mail Application Program Interface, such as Microsoft Exchange and outlook)
- VIM (Vendor Independent Mail, such as Lotus Notes)
- SMTP (Simple Mail Transfer Protocol). With SMTP you bypass the mail client and you have to specify outgoing mail account settings

You can set mail interfaces and the other email-system options in several ways:

- modify the SAS configuration file
- set them in the System Options window
- set them using an options statement

The SAS configuration file is normally located in the same directory where the SAS software is installed. For SAS 9.3 the default location is C:\Program Files\SASHome\SASFoundation\9.3\nls\en\sasv9.cfg. You can add the email lines in the upper part of the configuration file, for instance:

```
/* Set Mail options */  
-EMAILSYS SMTP  
-EMAILHOST "mail.mailserver.com"  
-EMAILID first.last@domain.com
```

The equivalent line in a SAS Options statement would be:

```
OPTIONS EMAILSYS=SMTP EMAILHOST="mail.mailserver.nl" EMAILID="first.last@domain.com";
```

The System options window is opened in the SAS Windowing environment via the menu: Tools -> Options -> System. Then choose the Communications group -> E-mail to see the current settings (Figure 1). To modify the settings, right-click on the option and open the modify window.

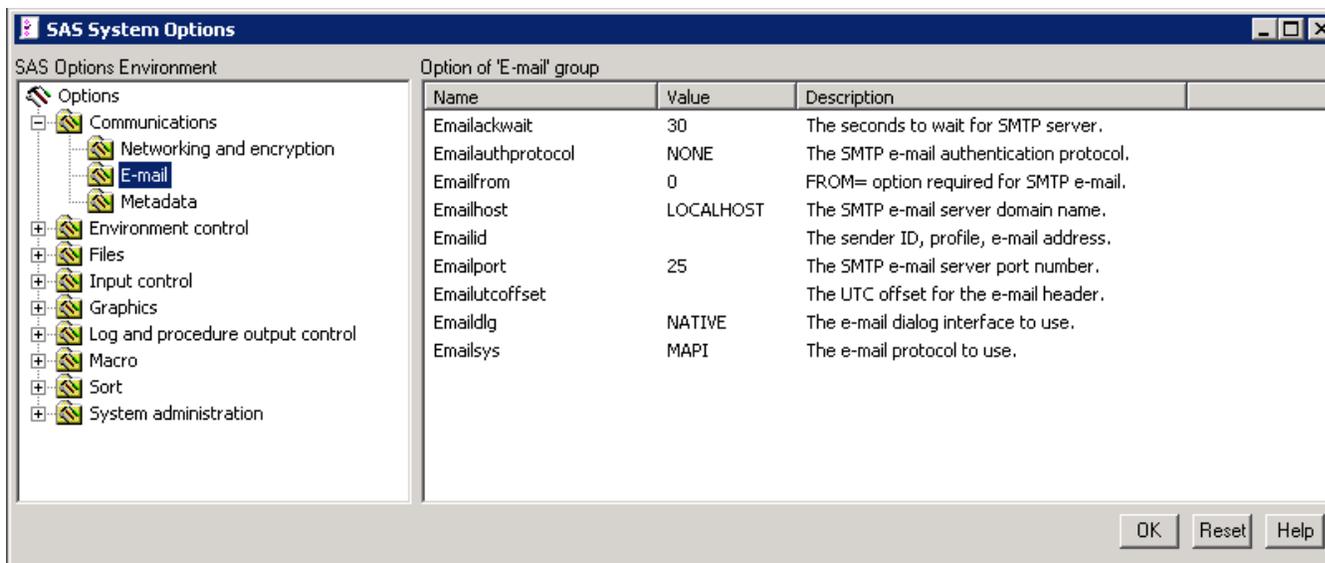


Figure 1. The System Options window with the EMAIL options

SETTING UP INTERFACE

- MAPI: The MAPI interface works with the mail system as installed on the computer where SAS runs. It retrieves the necessary information from that system, so you hardly need to specify anything. The only thing that is still missing is the EMAILID option in which you specify the ID that you want to use for sending the messages. This ID is case sensitive. The options you have to set are:

EMAILSYS MAPI

EMAILID "sender information"

- SMTP: SMTP bypasses the mail client that is present on the computer and contacts the SMTP server directly. So you have to specify information similar to the information you specify for your outgoing mail account in mail clients like Outlook. The options you have to set are:

EMAILSYS SMTP

EMAILHOST name of outgoing mail server

EMAILID "sender information"

EMAILPW "password to access mail server"

- For the VIM interface to work, you have to specify the EMAILSYS, EMAILID and EMAILPW options:

EMAILSYS VIM

EMAILID "User log on code"

EMAILPW "User password"

The sender information and password have to be enclosed in double quotes if they contain blanks.

For example: `OPTIONS EMAILSYS=SMTP EMAILHOST=mail.server.com EMAILID="Nirav Darji <fname.lname@domain.com>" EMAILPW=Abc@123;`

SAS STATEMENTS FOR EMAIL ACCESS METHOD

FILENAME STATEMENT

Enables you to send electronic mail programmatically from SAS.

Syntax

FILENAME *fileref* EMAIL <'address' ><*email-options*>;

Arguments

fileref

It is a valid file reference. The fileref is a name that is temporarily assigned to an external file or to a device type. Note that the fileref cannot exceed eight characters.

EMAIL

It specifies the EMAIL device type, which provides the access method that enables you to send electronic mail programmatically from SAS.

'address'

It is the e-mail address to which you want to send the message. You must enclose the address in quotation marks. Specifying an address as a FILENAME statement argument is optional if you specify the TO= e-mail option or the PUT statement !EM_TO! directive, which will override an address specification.

E-mail Options

You can use any of the following e-mail options in the FILENAME statement to specify attributes for the electronic message.

Note: You can also specify these options in the FILE statement. E-mail options that you specify in the FILE statement override any corresponding e-mail options that you specified in the FILENAME statement.

ATTACH='filename.ext' | ATTACH= ('filename.ext' attachment-options)

It specifies the physical name of the file or files to be attached to the message and any options to modify attachment specifications. Enclose the physical name in quotation marks. To attach more than one file, enclose the group of files in parentheses, enclose each file in quotation marks, and separate each with a space.

Here are examples:

```
attach="/u/userid/opinion.txt"
```

```
attach=('C:\Status\June2001.txt' 'C:\Status\July2001.txt')
```

The *attachment-options* include the following:

EXTENSION='extension'

It specifies a different file extension to be used for the specified attachment. You must enclose the value in quotation marks. This extension is used by the recipient's e-mail program for selecting the appropriate utility to use for displaying the attachment. For example, the following results in the attachment **home.html** being received as **index.htm**:

```
attach=("home.html" name="index" ext="htm")
```

Note: If you specify *extension=""*, the specified attachment will have no file extension.

Alias: EXT=

NAME=’filename’

It specifies a different name to be used for the specified attachment. You must enclose the value in quotation marks. For example, the following results in the attachment **home.html** being received as **index.html**:

```
attach=("home.html" name="index")
```

CONTENT_TYPE=’contenttype’

It specifies the content type for the attached file. You must enclose the value in quotation marks. If you do not specify a content type, SAS tries to determine the correct content type based on the filename. For example, if you do not specify a content type, a filename of **home.html** is sent with a content type of text/html.

Aliases: CT= and TYPE=

```
attach("home.html" name="index" type="text/html")
```

Default: If SAS cannot determine a content type based on the filename and extension, the default value is text/plain.

BCC=’bcc-address’

It specifies the recipient or recipients that you want to receive a blind copy of the electronic mail. Individuals that are listed in the **bcc** field will receive a copy of the e-mail. The BCC field does not appear in the e-mail header, so that these e-mail addresses cannot be viewed by other recipients.

If a BCC address contains more than one word, then enclose it in quotation marks. To specify more than one address, you must enclose the group of addresses in parentheses, enclose each address in quotation marks, and separate each address with a space. To specify a real name as well as an address, enclose the address in angle brackets (<>). Here are examples:

```
bcc="joe@site.com"
```

```
bcc=("joe@site.com" "jane@home.net")
```

```
bcc="Joe Smith <joe@site.com>"
```

CC=’cc-address’

It specifies the recipient or recipients to receive a copy of the e-mail message. You must enclose an address in quotation marks. To specify more than one address, enclose the group of addresses in parentheses, enclose each address in quotation marks, and separate each address with a space. To specify a real name as well as an address, enclose the address in angle brackets (<>). Here are examples:

```
cc='joe@site.com'
```

```
cc=("joe@site.com" "jane@home.net")
```

```
cc="Joe Smith <joe@site.com>"
```

FROM=’from-address’

It specifies the e-mail address of the author of the message that is being sent. The default value for FROM= is the e-mail address of the user who is running SAS. For example, specify this option when the person who is sending the message from SAS is not the author. You must enclose an address in quotation marks. You can specify only one e-mail address. To specify the author’s real name along with the address, enclose the address in angle brackets (<>). Here are examples:

```
from='martin@home.com'
```

```
from="Brad Martin <martin@home.com>"
```

Requirement: The FROM option is required if the EMAILFROM system option is set.

IMPORTANCE= 'LOW' | 'NORMAL' | 'HIGH'

It specifies the priority of the e-mail message. You must enclose the value in quotation marks. Here are examples:

```
filename inventory email 'name@mycompany.com' importance='high';
```

Default: NORMAL

REPLYTO='replyto-address'

It specifies the e-mail address(es) for who will receive replies. You must enclose an address in quotation marks. To specify more than one address, enclose the group of addresses in parentheses, enclose each address in quotation marks, and separate each address with a space. To specify a real name along with an address, enclose the address in angle brackets (< >). Here are examples:

```
replyto='hiroshi@home.com'  
replyto=('hiroshi@home.com' 'akiko@site.com')  
replyto="Hiroshi Mori <mori@site.com>"
```

SUBJECT=subject

It specifies the subject of the message. If the subject contains special characters or more than one word (that is, it contains at least one blank space), you must enclose the text in quotation marks. Here are examples:

```
subject=Sales  
subject="June Sales Report"
```

Note: If you do not enclose a one-word subject in quotation marks, it is converted to uppercase.

TO='to-address'

It specifies the primary recipient or recipients of the e-mail message. You must enclose the address in quotation marks. To specify more than one address, enclose the group of addresses in parentheses, enclose each address in quotation marks, and separate each address with a space. To specify a real name as well as an address, enclose the address in angle brackets (< >). Here are examples:

```
to='joe@site.com'  
to=("joe@site.com" "jane@home.net")  
to="Joe Smith <joe@site.com>"
```

Tip: Specifying TO= overrides the 'address' argument.

PUT STATEMENT

In the DATA step, after using the FILE statement to define your e-mail fileref as the output destination, use PUT statements to define the body of the message. For example,

```
filename mymail email 'martin@site.com' subject='Sending Email';  
data _null_;  
  file mymail;  
  put 'Hi';  
  put 'This message is sent from SAS...';  
run;
```

You can also use PUT statements to specify e-mail directives that override the attributes of your message (the e-mail options like TO=, CC=, SUBJECT=, CONTENT_TYPE=, ATTACH=), or to perform actions such as send, end abnormally, or start a new message. Specify only one directive in each PUT statement; each PUT statement can contain only the text that is associated with the directive that it specifies.

The directives that change the attributes of a message are as follows:

'!EM_ATTACH! *filename.ext* | ATTACH=(*filename.ext* *attachment-options*)'

It replaces the physical name of the file or files to be attached to the message and any options to modify attachment specifications. The physical name is the name that is recognized by the operating environment. The directive must be enclosed in quotation marks, and the physical name must be enclosed in quotation marks. To attach more than one file, enclose the group of files in parentheses, enclose each file in quotation marks, and separate each with a space. Here are examples:

```
put '!em_attach! /u/userid/opinion.txt';  
put '!em_attach! ("C:\Status\June2001.txt" "C:\Status\July2001.txt");
```

The *attachment-options* are same as for FILENAME statement like EXTENSION, NAME, CONTENT_TYPE etc.

'!EM_BCC! *bcc-address*'

It replaces the current blind copied recipient address(es) with *addresses*. These recipients are not visible to the recipients in the !EM_TO! or !EM_CC! addresses. If you want to specify more than one address, then you must enclose the group of addresses in parentheses, enclose each address in quotation marks, and separate each address with a space. To specify real names along with addresses, enclose the address in angle brackets (< >). Here are examples:

```
put '!em_bcc! joe@site.com';  
put '!em_bcc! ("joe@site.com" "jane@home.net");  
put '!em_bcc! Joe Smith <joe@site.com>';
```

'!EM_CC! *cc-address*'

It replaces the current copied recipient address(es). The directive must be enclosed in quotation marks. To specify more than one address, enclose the group of addresses in parentheses, enclose each address in quotation marks, and separate each address with a space. To specify real names along with addresses, enclose the address in angle brackets (< >). Here are examples:

```
put '!em_cc! joe@site.com';  
put '!em_cc! ("joe@site.com" "jane@home.com");  
put '!em_cc! Joe Smith <joe@site.com>';
```

'!EM_FROM! *from-address*'

It replaces the current address of the author of the message being sent, which could be either the default or the one specified by the FROM= e-mail option. The directive must be enclosed in quotation marks. You can specify only one e-mail address. To specify the author's real name along with the address, enclose the address in angle brackets (< >). Here are examples:

```
put '!em_from! martin@home.com';  
put '!em_from! Brad Martin <martin@home.com>';
```

'!EM_IMPORTANCE! LOW | NORMAL | HIGH'

It specifies the priority of the e-mail message. The directive must be enclosed in quotation marks. Here are examples:

```
put '!em_importance! high';
```

Default: NORMAL

'!EM_REPLYTO! *replyto-address*'

It replaces the current address(es) of who will receive replies. The directive must be enclosed in quotation marks. To specify more than one address, enclose the group of addresses in parentheses, enclose each address in quotation marks, and separate each address with a space. To specify a real name along with an address, enclose the address in angle brackets (< >). Here are examples:

```
put '!em_replyto! hiroshi@home.com';  
put '!em_replyto! ("hiroshi@home.com" "akiko@site.com")';  
put '!em_replyto! Hiroshi Mori <mori@site.com>';
```

'!EM_SUBJECT! *subject*'

It replaces the current subject of the message. The directive must be enclosed in quotation marks. If the subject contains special characters or more than one word (that is, it contains at least one blank space), you must enclose the text in quotation marks. Here are examples:

```
put '!em_subject! Sales';  
put '!em_subject! "June Sales Report"';
```

'!EM_TO! *to-address*'

It replaces the current primary recipient address(es). The directive must be enclosed in quotation marks. To specify more than one address, enclose the group of addresses in parentheses, enclose each address in quotation marks, and separate each address with a space. To specify a real name along with an address, enclose the address in angle brackets (< >). Here are examples:

```
put '!em_to! joe@site.com';  
put '!em_to! ("joe@site.com" "jane@home.net")';  
put '!em_to! Joe Smith <joe@site.com>';
```

Tip: Specifying !EM_TO! overrides the '*address*' argument and the TO= e-mail option.

Here are the directives that perform actions:

'!EM_SEND!'

It sends the message with the current attributes. By default, SAS sends a message when the fileref is closed. The fileref closes when the next FILE statement is encountered or the DATA step ends. If you use this directive, SAS sends the message when it encounters the directive, and again at the end of the DATA step.

This directive is useful for writing DATA step programs that conditionally send messages or use a loop to send multiple messages.

'!EM_ABORT!'

It abnormally end the current message. You can use this directive to stop SAS from automatically sending the message at the end of the DATA step. By default, SAS sends a message for each FILE statement.

'!EM_NEWMSG!'

It clears all attributes of the current message that were set using PUT statement directives.

EXAMPLES

EXAMPLE 1: SENDING E-MAIL WITH AN ATTACHMENT USING A DATA STEP

In order to share a copy of your SAS configuration file with another user, you could send it by submitting the following program. The e-mail options are specified in the FILENAME statement:

```
filename mymail email "JBrown@domain.com"
  subject="My SAS Configuration File"
  attach="/u/sas/sasv8.cfg";
data _null_;
  file mymail;
  put 'Jim,';
  put 'This is my SAS configuration file.';
  put 'I think you might like the';
  put 'new options I added.';
run;
```

The following program sends a message and two file attachments to multiple recipients. For this example, the e-mail options are specified in the FILE statement instead of the FILENAME statement.

```
filename outbox email "ron@domain.com";
data _null_;
  file outbox
    to=("ron@domain.com" "humberto@domain.com")
    /* Overrides value in */
    /* filename statement */
    cc=("miguel@domain.com" "loren@domain.com")
    subject="My SAS Output"
    attach=("C:\sas\results.out" "C:\sas\code.sas")
  ;
  put 'Folks,';
  put 'Attached is my output from the SAS';
  put 'program I ran last night.';
  put 'It worked great!';
run;
```

EXAMPLE 2: USING CONDITIONAL LOGIC IN A DATA STEP

You can use conditional logic in a DATA step in order to send multiple messages and control which recipients get which message. For example, in order to send customized reports to members of two different departments, the following program produces an e-mail message and attachments that are dependent on the department to which the recipient belongs. In the program, the following occurs:

1. In the first PUT statement, the !EM_TO! directive assigns the TO attribute.
2. The second PUT statement assigns the SUBJECT attribute using the !EM_SUBJECT! directive.
3. The !EM_SEND! directive sends the message.
4. The !EM_NEWMSG! directive clears the message attributes, which must be used to clear message attributes between recipients.
5. The !EM_ABORT! directive abnormally ends the message before the RUN statement causes it to be sent again. The !EM_ABORT! directive prevents the message from being automatically sent at the end of the DATA step.

```
filename reports email "Jim.Smith@work.com";
```

```
data _null_;  
  file reports;  
  length name dept $ 21;  
  input name dept;  
  put '!EM_TO! ' name;  
  put '!EM_SUBJECT! Report for ' dept;  
  put name ',';  
  put 'Here is the latest report for ' dept '.' ;  
  if dept='marketing' then  
    put '!EM_ATTACH! c:\mktrept.txt';  
  else /* ATTACH the appropriate report */  
    put '!EM_ATTACH! c:\devrept.txt';  
  put '!EM_SEND!';  
  put '!EM_NEWMSG!';  
  put '!EM_ABORT!';  
  
  datalines;  
  Susan marketing  
  Peter marketing  
  Alma development  
  Andre development  
  ;  
run;
```

EXAMPLE 3: SENDING PROCEDURE OUTPUT IN E-MAIL

You can use e-mail to send procedure output. This example illustrates how to send ODS HTML in the body of an e-mail message. Note that ODS HTML procedure output must be sent with the RECORD_SEPARATOR (RS) option set to NONE.

```
filename outbox email  
  to='susan@domain.com'  
  type='text/html'
```

```
        subject='Temperature Conversions';
data temperatures;
    do centigrade = -40 to 100 by 10;
        fahrenheit = centigrade*9/5+32;
        output;
    end;
run;
ods html
    body=outbox /* Mail it! */
rs=none;
title 'Centigrade to Fahrenheit Conversion Table';
proc print;
    id centigrade;
    var fahrenheit;
run;
ods html close;
```

EXAMPLE 4: CREATING AND E-MAILING AN IMAGE

The following example illustrates how to create a GIF image and send it from SAS in an e-mail message:

```
filename gsasfile email
    to='Jim@acme.com'
    type='image/gif'
    subject="SAS/GRAPH Output";
goptions dev=gif gsfname=gsasfile;
proc gtestit pic=1;
run;
```

EXAMPLE 5: SENDING MULTIPLE EMAILS

You can send more than one message simultaneously for different addressees, you may specify more FILENAME statements and direct the PUT statement to either of them with different FILE statements. All messages will be sent at the end of the data step. Program 3 shows the principle:

```
filename mailbox1 email;
filename mailbox2 email;
filename mailbox3 email;
data _null_;
    file mailbox1 to='erik <erik_tilanus@domain.com>'
        subject='first addressee';
    put "hello mailbox1";
    file mailbox2 to='erik.tilanus@domain.com'
```

```
    subject='second addressee';  
    put "hello mailbox2 ";  
    file mailbox3 to='synchrona@domain.com'  
    subject='third addressee';  
    put "hello mailbox3 ";  
run;
```

CONCLUSION

SAS EMAIL engine can be very useful for automation. It can help teams who works in shifts or works in different time zones in different countries by sending emails using SAS. It is an easy but very powerful tool.

REFERENCES

SAS 9.3 Product Documentation: <http://support.sas.com/documentation/93/>

ACKNOWLEDGMENTS

I would like to thank my friends and colleagues at GCE Solutions, for motivating me to write this paper, documenting my work on SAS and also inspiring me and to present at the conference.

CONTACT INFORMATION

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

Name: Nirav Darji
Enterprise: GCE Solutions
Work Phone: +1-309-807-5879 / +91-11-41069686
Email: nirav.darji@gcesolutions.com / niravmdarji@gmail.com
Website: <https://gcesolutions.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.