

Paper 023-2013

Using Mail Functionality in SAS®

Erik W. Tilanus, Synchrona, the Netherlands

ABSTRACT

The DATA step can write information to virtually any destination. You are probably familiar with writing to SAS data sets and external files. But also email can function as a destination. The paper will discuss how to configure the email features in the system options and share practical hints how to use them. Then we will proceed with sending a simple email from the DATA step, with or without attachments. After that we will use extensions to the standard PUT statement to support the email facility to send personalized mass-mailings. Finally, we will show how to send procedure output that has been created using ODS.

INTRODUCTION

The DATA Step is the core of the data processing facilities in SAS. It is used to read information from virtually any source. It is used to process the data any way you like and eventually to write the processed data to virtually any destination.

One of those destinations may be an email server, meaning that you can send emails directly from the DATA Step. In order to do that, you first have to define the email environment that SAS will use. This environment is defined in a number of system options.

Once the environment is set up, sending email is like writing information to an external destination, using FILENAME, FILE and PUT statements. It can be as simple as sending a mail with a fixed content to a fixed mail address, but it is also possible to send personalized mails to a whole list of addressees.

SETTING UP TO SEND E-MAIL

SAS supports three interface methods to send mail:

MAPI: "Messaging Application Program Interface". This is the default and makes use of the mail client on your computer, e.g. Microsoft Exchange or Outlook or Mozilla Thunderbird.

SMTP: "Simple Mail Transfer Protocol". With SMTP you bypass the mail client and you have to specify outgoing mail account settings, comparable to specifying the outgoing mail settings in Outlook or Thunderbird.

VIM: "Vendor Independent Mail". VIM can be used in combination with Lotus Notes or cc:Mail.

The method is selected with the EMAILSYS option: EMAILSYS=MAPI|SMTP|VIM

You can set this option 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.

Either way: the options you have to set are the same.

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.nl"
-EMAILID "erik.tilanus@planet.nl"
```

The equivalent line in a SAS Options statement would be:

```
OPTIONS EMAILSYS=SMTP EMAILHOST="mail.mailserver.nl"
        EMAILID="erik.tilanus@planet.nl";
```

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 2).

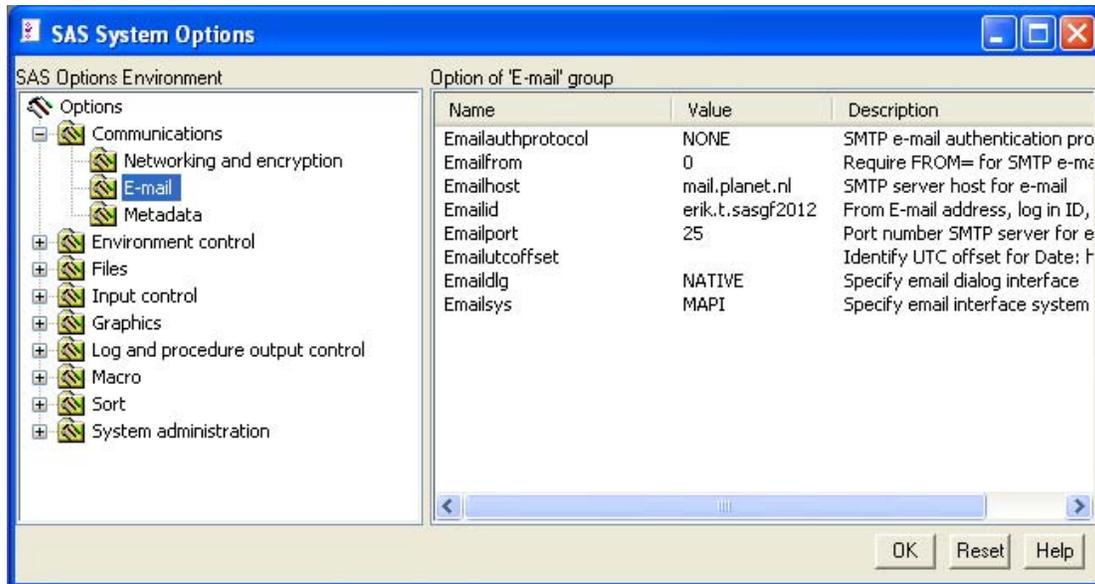


Figure 1: The System Options window with the EMAIL options.

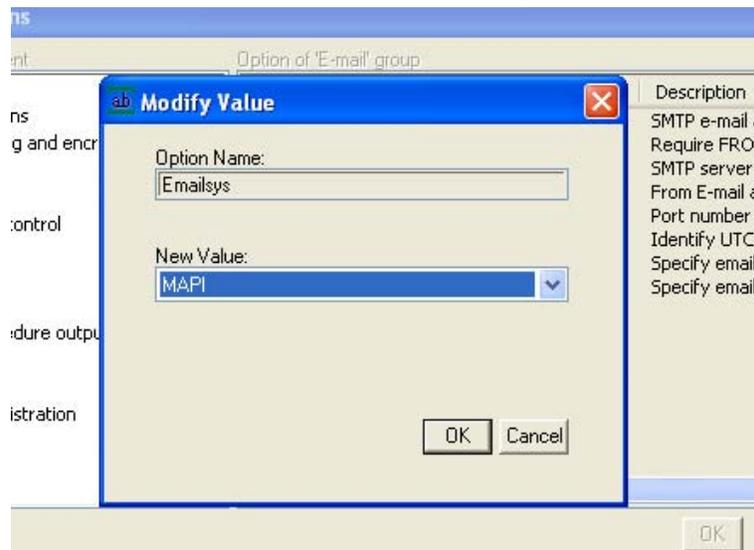


Figure 2: Right-Click on an option to modify it.

SETTING UP THE MAPI INTERFACE

MAPI is the default method to be used by SAS. 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. Just verify that the system option EMAILSYS has the value MAPI and you are almost ready to go.

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, which is normally your mail address up to the @ sign. Be aware: this ID is case sensitive, as can be seen in the partial log presented in Listing 1.

```

1  options emailsys=MAPI emailid='erik_tilanus';
2  filename mailbox email;
3  data _null_;
4  file mailbox to='erik.tilanus@planet.nl' subject='MAPI test';
5  put "Hi";
6  run;

```

NOTE: The file MAILBOX is:
E-Mail Access Device

```

Message sent
  To:          erik.tilanus@planet.nl
  Cc:
  Bcc:
  Subject:    MAPI test
  Attachments:

```

NOTE: 1 record was written to the file MAILBOX.
The minimum record length was 2.
The maximum record length was 2.

NOTE: DATA statement used (Total process time):
real time 0.06 seconds
cpu time 0.04 seconds

```

7  options emailsys=MAPI emailid='Erik_tilanus';
8  filename mailbox email;
9  data _null_;
10 file mailbox to='erik.tilanus@planet.nl' subject='MAPI test';
11 put "Hi";
12 run;

```

ERROR: Login failure.

NOTE: The SAS System stopped processing this step because of errors.

NOTE: DATA statement used (Total process time):
real time 0.01 seconds
cpu time 0.00 seconds

Listing 1: SAS LOG messages when using the MAPI interface.

SETTING UP THE SMTP INTERFACE

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 or Thunderbird. So look under "preferences" or "outgoing mail settings" or a similar heading in your mail client to find the right settings. The options you have to set are

EMAILSYS	SMTP
EMAILHOST	name of outgoing mail server
EMAILID	"sender information"
EMAILPW	"password to access mail server"

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

For example: OPTIONS EMAILSYS=SMTP EMAILHOST=mail.planet.nl EMAILID="Erik Tilanus
<erik.tilanus@planet.nl>" EMAILPW=etila999;

There are two more options that can be set: EMAILPORT and EMAILAUTHPROTOCOL. EMAILPORT defines the port to which the SMTP server listens. The default port is 25, which will be correct in most cases.

EMAILAUTHPROTOCOL defines whether you need a login sequence for the outgoing mail server.

SETTING UP THE VIM INTERFACE

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
```

By the way, if you are sending mails from the Display Manager menu (File → Send Mail...) SAS will use either the MAPI or the VIM interface and not the SMTP interface.

GMAIL AND HOTMAIL

GMAIL and HOTMAIL require a secure link between the mail client and the SMTP server. That is currently not supported in the SMTP interface. So if you want to send mails via those mail networks, you have to use the MAPI interface and let the mail client on your computer take care of the secure link.

OUTLOOK POP-UP

Sometimes Outlook is configured that it does not take messages sent via the MAPI interface for granted and comes with a pop-up to ask permission to send the mail, which can of course be quite annoying.

This is something you cannot change within SAS, but you have to change the Windows registry for it. After taking the necessary back-ups, look for the following registry entry:

```
HKEY_CURRENT_USER\Software\Policies\Microsoft\Office\14.0\Outlook\Security
```

If it does not yet exist, create it. If it exists look for the following elements:

```
PromptSimpleMAPISend
PromptSimpleMAPINameResolve
PromptSimpleMAPIOpenMessage
```

If they do not exist, go to the Edit menu and choose NEW - DWORD Value. Then create the elements.

After locating or creating the elements right-click on them and modify their value to 2 (=auto-approve).

SENDING A MAIL MESSAGE FROM A DATA STEP

Once you have set up the system options, you use a FILENAME statement to define the output destination, a FILE statement to link your PUT statements to the right output destination and PUT statements to create the message. :

```
FILENAME Mailbox EMAIL 'Erik.Tilanus@planet.nl'
      Subject='Test Mail message';
DATA _NULL_;
FILE Mailbox;
PUT "Hello";
PUT "This is a test message from the DATA step";
RUN;
```

Program 1: Basic DATA step to send mail.

In Program 1 the addressee and the subject were included in the FILENAME statement. You don't need to do that. As an alternative you can also specify them in the FILE statement as in Program 2. If they are present in both, the information in the FILE statement will prevail.

In the same way you can enter also copy-readers, by specifying CC=mail-address or BCC=mail-address and add attachments (ATTACH=). The latter will be discussed in a later paragraph.

To include a display name you place the whole address between quotes and the address part between < and >, like in TO='Erik <Erik.Tilanus@planet.nl>'.

If you want to send a mail to more than one address, you list all addresses (each between quotes) within parentheses.

```

FILENAME Mailbox EMAIL;
DATA _NULL_;
FILE Mailbox TO='Erik_Tilanus@hotmail.com'
           SUBJECT='Test Mail message';
PUT "Hello";
PUT "This is a message from the DATA step";
RUN;

```

Program 2: As Program 1, but with all mail data moved from FILENAME to FILE statement.

If you are using the SMTP interface, you can overrule the default sender and specify a separate reply address. Use the FROM option and the REPLYTO option to accomplish this. Note that REPLYTO does not work with the MAPI interface, since it uses the information as programmed in the mail account. If that contains a separate reply-address, fine, it will be used, but you cannot overrule it.

All these options are put together in the following FILE statement:

```

FILE MailBox TO=('Erik <Erik@synchrona.nl>'
               'Myself <Erik.Tilanus@planet.nl>')
             CC=('Synchrona@planet.nl'
               'Erik_Tilanus@hotmail.com')
             BCC='Anneke.Tilanus@planet.nl'
             FROM='My Business <Synchrona@planet.nl>'
             REPLYTO='My anonymous address <villavinkeveen@hotmail.com>'
             SUBJECT='Test Mail message to group and other options';

```

Replacing the FILE statement in Program 2 with this one results in the message as displayed in Figure 3.

If your outgoing mail server is not picky, you can send anonymous mails this way. To demonstrate this I set the EMAILID to "nobody@nowhere.com". Then I used Program 2 to send an e-mail. Figure 3 shows the mail as received by me. I also checked what happened if I leave EMAILID blank. Then SAS will use the Windows user and system name as the sender information.

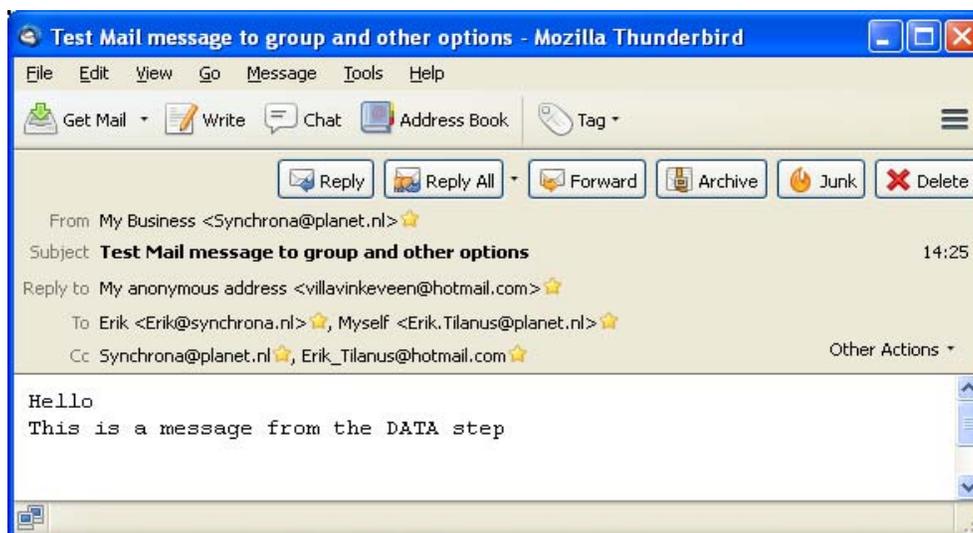


Figure 3: Mail received from Program 2 with the extended FILE statement.

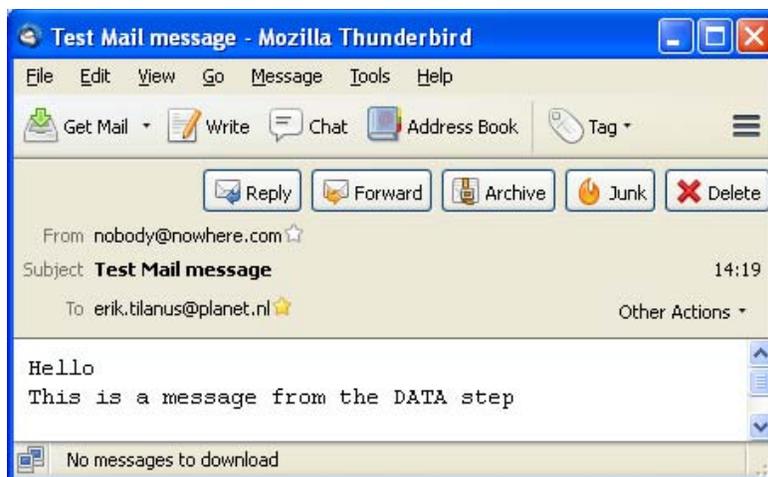


Figure 4: Anonymous mail.

If you want to build 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.

Program 3: Sending more mails simultaneously.

```
FILENAME MailBox1 EMAIL;
FILENAME MailBox2 EMAIL;
FILENAME MailBox3 EMAIL;

DATA _NULL_;
FILE MailBox1 TO='Erik <Erik_Tilanus@compuserve.com>'
             SUBJECT='First addressee';
PUT "Hello Mailbox1";
FILE MailBox2 TO='erik.tilanus@planet.nl'
             SUBJECT='Second addressee';
PUT "Hello Mailbox2 ";
FILE MailBox3 TO='synchrona@planet.nl'
             SUBJECT='Third addressee';
PUT "Hello Mailbox3 ";
RUN;
```

SENDING ATTACHMENTS

Including attachments is also rather straightforward: specify the attachment in either the FILENAME or FILE statement with the keyword ATTACH, like:

```
FILENAME MailBox EMAIL ATTACH='C:\SASUtil\Tips.doc';
or
FILE Mailbox ATTACH='C:\SASUtil\Tips.doc';
```

If you want to specify the attachment under program control, you can use the special option in the PUT statement:

```
PUT '!EM_ATTACH!' AttachInfo;
```

where AttachInfo is either a character variable that contains the full path and name of the file that should be attached or the full path and name between quotes. More about this special PUT directive and other similar constructions are discussed below in the paragraph "Personalized bulk mail".

SENDING OUTPUT

Output can be mailed in two ways: either by first creating an output file then mailing it as an attachment, or specify a MAIL destination in an ODS HTML statement. The first method is more universal, so lets do that first.

CREATING AN OUTPUT FILE AND SENDING IT AS AN ATTACHMENT

The principle is simple: write the (procedure-)output to a file and then specify that file as an attachment in a subsequent DATA _NULL_ step that sends the mail. If you don't need a permanent copy of the output, you can store it in a temporary file. Don't do that with a FILENAME ... TEMP; since such file does not have a file extension and therefore the output file type is not recognized by the receiver of the mail. The solution is to create a **permanent** file at a **temporary** location, for instance the WORK library.

Program 4 shows how this works. The program consists of two parts: the first part creates the output, the second part sends it. In this example we used a DATA step to produce the output, but it works off course the same with procedure output. The enclosing ODS statements specify that the report is to be created in PDF format and filed at the address in the FILENAME statement.

In the FILENAME statement you see a not too familiar construction: %SYSFUNC(GETOPTION(WORK)). The %SYSFUNC macro lets you use DATA step functions outside a DATA step. In this case we use the GETOPTION function to get the address of the current WORK library. In combination with the filename (testlist.pdf) we have the full path of the output file.

The second DATA step picks up the file name again, by opening the file with the FOPEN function and getting the full name using the FINFO function. Thus it assigns the full file name to the variable Attachment and attaches the file to the mail with the special PUT statement that follows.

```
FILENAME MailBox EMAIL 'Erik.Tilanus@planet.nl'
      SUBJECT='Mail message with PDF attachment';
FILENAME pdffile "%SYSFUNC(GETOPTION(WORK))\testlist.pdf";
ODS LISTING CLOSE;
ODS PDF FILE=pdffile;
DATA _NULL_;
FILE PRINT;
PUT "This is my report";
TITLE 'Test report in PDF format';
RUN;
ODS PDF CLOSE;
ODS LISTING;

DATA _NULL_;
attachment=FINFO(FOPEN('pdffile'),'FileName');
FILE MailBox;
PUT "attached you find the report";
PUT '!EM_ATTACH!' attachment;
RUN;
```

Program 4: Creating a report in a DATA step and sending it as attachment

SENDING HTML AND RTF OUTPUT DIRECTLY FROM ODS

HTML and RTF output can be sent also directly using ODS. It does not work with other destinations, like PDF. The simple reason for this is that the output is not sent as an attachment, like in Program 4, but in the body of the message.

Most mail clients will recognize HTML information and present it correctly when opening the e-mail. RTF output however would display as a text message showing all the RTF commands. To display the output correctly, you will have to save the contents of the message as a text file with extension RTF and then reopen it as RTF document with for instance MS Word or OpenOffice Writer.

Program 5 shows an example. We prepare a PROC TABULATE report with sales data for two years and the index figures of the second year over the first year, including some traffic lighting. Figure 5 shows the report as it is received by mail.

```

FILENAME mail EMAIL TO="erik.tilanus@planet.nl"
SUBJECT="HTML TABULATE OUTPUT" CONTENT_TYPE="text/html";
ODS HTML BODY=mail;
TITLE 'Turnover 2009/10 over 2008/09';
PROC TABULATE DATA=SalesData
STYLE=[background=black foreground=white fontsize=4];
CLASS ProductGroup Product/STYLE=[background=black foreground=white
fontsize=4];
CLASSLEV Product/STYLE=[background=yellow foreground=black fontsize=4
fontstyle=italic];
CLASSLEV Productgroup/STYLE=[fontsize=4];
VAR Turnover_CY Turnover_PY/STYLE=[background=brown foreground=yellow
fontsize=3];

TABLE Productgroup*Product,
Turnover_PY*SUM=' ' Turnover_CY*SUM=' ' Turnover_CY=' '*
{STYLE=[background=index. fontsize=3]}*PCTSUM<Turnover_PY>/
RTS=32;
LABEL Turnover_PY = 'Turnover 2008/09'
Turnover_CY = 'Turnover 2009/10'
productgroup = 'Product Group';
KEYLABEL PCTSUM='Index 2008/09';
KEYWORD PCTSUM/STYLE=[background=lightblue fontsize=4];
RUN;
ODS HTML CLOSE;

```

Program 5: Sending procedure output directly by mail (HTML or RTF)

HTML TABULATE OUTPUT - Mozilla Thunderbird

From: Erik.Tilanus@planet.nl
Subject: HTML TABULATE OUTPUT
To: erik.tilanus@planet.nl

Turnover 2009/10 over 2008/09

Product Group	Product	Turnover 2008/09	Turnover 2009/10	Index 2008/09
Alcoholic	Beer	2205.00	2754.00	124.90
	Wine	1569.00	2186.00	139.32
Fruit juices	Apple juice	286.00	250.00	87.41
	Orange Juice	345.00	356.00	103.19
	Pineapple juice	124.00	112.00	90.32
Soda's	Cola	686.00	813.00	118.51
	Tonic	560.00	713.00	127.32

Figure 5: HTML output directly from ODS.

SENDING A MESSAGE UNDER SOFTWARE CONTROL

In the previous paragraphs the addressing of e-mails and the setting of subject and so on were handled in either the FILENAME or the FILE statement.

As an alternative you can bring these settings also under software control in the DATA step using special directives in the PUT statement.

The general form is

```
PUT '!EM_xxx!' specification;
```

where xxx stands for some directive. The !EM_xxx! directive should be the first element on any output line, so normally it is the first element in a PUT statement, but it can also be the first element after a "new line" directive (/) in the PUT statement. There cannot be two directives on the same line. The second would be interpreted as text.

Table 1 shows the possible directives. The examples in this table show quoted text as directive and specification. However that is not required. You could as well use character variables containing the appropriate information. It's a PUT statement, remember! When specifying more addresses, each address should be between quotes and the total string should be between quotes, as you can see in the !EM_CC! Example.

Directive	Specification	Example
!EM_TO!	Mail address, as in FILENAME TO=	PUT '!EM_TO!' "erik.tilanus@planet.nl";
!EM_CC!	Mail address, as in FILENAME CC=	PUT '!EM_CC!' "'erik.tilanus@planet.nl' 'erik_tilanus@hotmail.com'";
!EM_BCC!	Mail address, as in FILENAME BCC=	PUT '!EM_BCC!' "synchrona@planet.nl";
!EM_SUBJECT!	Subject for message between quotes	PUT '!EM_SUBJECT!' "My SAS tips";
!EM_ATTACH!	Full file address, as in FILENAME ATTACH=	PUT '!EM_ATTACH!' "C:\SASUtil\Tips.doc";
!EM_SEND!	None	PUT '!EM_SEND!';
!EM_NEWMSG!	None	PUT '!EM_NEWMSG!';
!EM_ABORT!	None	PUT '!EM_ABORT!';

Table 1: The special PUT statements to manage sending of mail messages.

The !EM_NEWMSG! directive clears all previous information and lets you start a new message. If you leave it out, every new message will build upon the previous one, just adding the information rather than replacing it.

The !EM_ABORT! prevents sending a message at termination of the DATA step. If you leave it out and you issued a !EM_NEWMSG! directive before, an empty message will be sent to the address in the FILENAME or FILE statement.

Program 6 shows an example how the directives can be used and Figure 6 shows the result.

```
filename mailbox email;
data _null_;
FILE MailBox;
My_CC = "'Synchrona@planet.nl' 'Erik_Tilanus@hotmail.com'";
PUT '!EM_TO!' 'Erik <Erik@synchrona.nl>';
PUT '!EM_CC!' My_CC /
    '!EM_SUBJECT!' 'Test message with !EM_xxx! directives' /
    "Hello" /
    "This time everything is managed within the DATA step";
run;
```

Program 6: Sending a message under software control.



Figure 6: The mail that is created with the !EM_XXX! directives.

PERSONALIZED BULK MAIL

The !EM_XXX! directives open up a totally different use of the mail functionality as well. Any time you process the !EM_SEND! directive a message is sent. So if you want to send mails to a whole mailing list that is in a SAS data set, you simply read that mailing list with a SET statement, then you build the message in the DATA step and finalize with an !EM_SEND! Directive.

However there are two things you have to worry about: the speed of your outgoing connection and the policies of the outgoing mail server. Your computer can probably spit out more messages than your line can handle and the mail server may conclude that you are spamming if you are spitting out too many messages too fast! (I have had that once when I was using a HOTMAIL address to send a PDF-newsletter to a 500 addresses mailing list!)

EXAMPLE BULK MAIL

Let us take the example of the newsletter. You want to address every person individually and let us assume that you want to include a reminder to renew the subscription if they did not do so for the last 12 months. The address list and the last renewal date are in a SAS data set (Figure 7).

Obs	Given_name	Surname	MailAddress	Member Since	Renewal Date
1	John	Smith	JSMITH@pops.com	12DEC2005	21DEC2012
2	Peter	North	PN@Northcorp.com	01APR2007	01JAN2012
3	Mary	Sunface	MarySunface@hotmail.com	15JUN2012	.
4	Georg	Amesbury	amesbury.g@gmail.com	01JAN2012	.

Figure 7: First four observations in the mailing data set.

My preference is that the text of the mail is not hard-coded in the mailing program itself, rather I would use a separate text file that I can read in. This text file can contain place holders where you have to include names and so on. Figure 8 Shows the mail message we want to send. The paragraph starting with #####3 should be included only when a renewal is overdue, which is for the second and fourth observation in Figure 7.

Program 7 shows the related program.

The program starts with reading the message text into an temporary array. That way it readily available whenever we build a message.

Then follows a speed throttle. To avoid overload situations and spamming suspicion we build a pause after every three messages and wait for 5 seconds. Depending on you line speed and the size of possible attachments you may need to adjust these values. In practice I have been using 1 to 5 messages per burst and 3 – 10 seconds sleep time. It never hurts when you are slowing down, except that it takes longer to send a long list of addresses.

Dear #####1 #####2,

Attached you will find this month's newsletter.

#####3It has been #####4 months since you #####5 for our newsletter. We hope you still enjoy it. To make sure that we only send the newsletter to those who want to receive it, we request you to renew your subscription by sending us an email.

#####3

We wish you pleasant reading hours.

The publisher

Figure 8: The text of the mail message with place holders.

After the throttle starts the construction of the message. First we fill the addressee, the sender, the subject and the link to the attachment. Then follows the compilation of the message body. We read a line from the text array and look for place holders. If they are found they are replaced by the real contents.

Once all place holders are solved the loop is left and the resulting line is sent to the mail destination with a PUT statement.

Place holder 3 is a special one as outlined above: if renewal is not needed, those paragraphs should not be included. To accomplish that we replace the line by \$\$\$\$ in case a renewal is not needed. Thus all other place holders are eliminated and the line is skipped at the PUT statement.

Listing 2 Shows the messages in the SAS Log and Figure 9 shows two of the mails as they are sent.

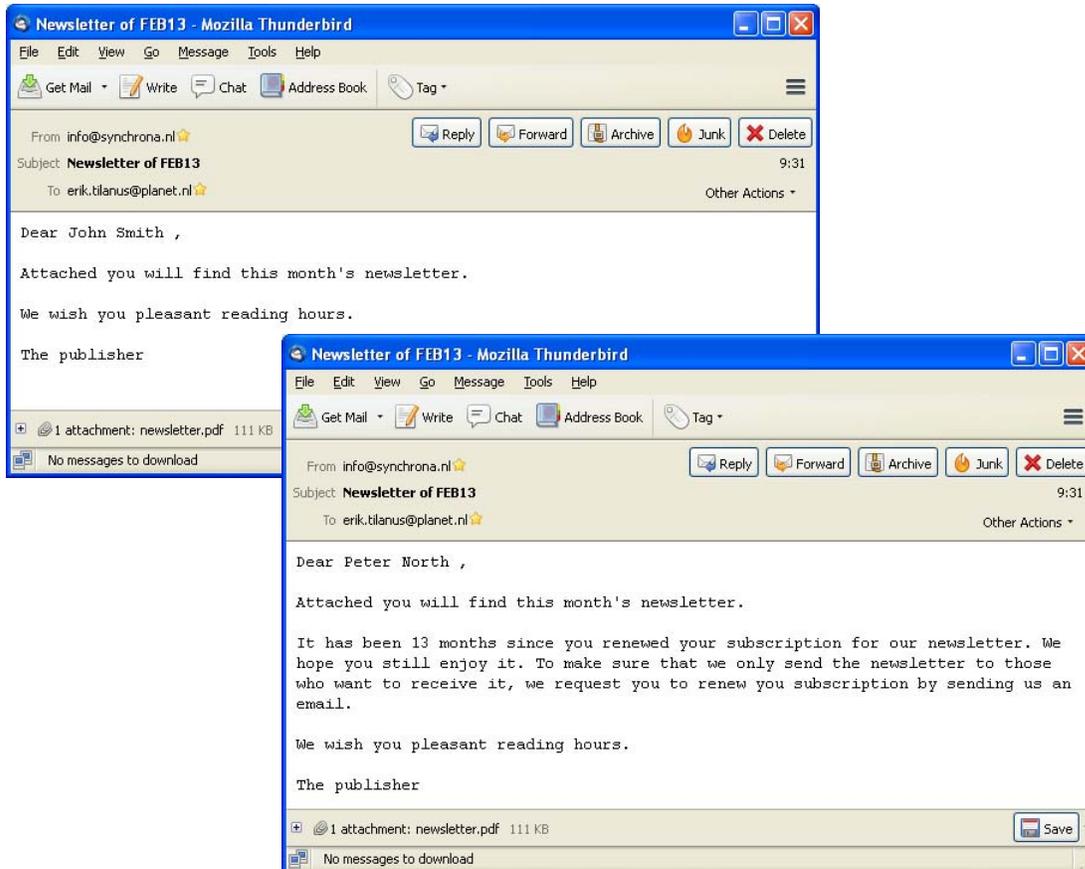


Figure 9: Personalized mass-mailing result.

```

FILENAME MailBox EMAIL LRECL=3200 CONTENT_TYPE='plain/text';
FILENAME MailMsg 'E:\SASforum\2013\mailtext.txt';
DATA _NULL_;
RETAIN paragraph_count;
LENGTH subject $100 line $3200 substitute $100;
* read the message text into a temporary array;
ARRAY mailtext {100} $3200 _TEMPORARY_;
IF _n_=1 THEN DO paragraph_count = 1 TO 100;
  INFILE MailMsg LENGTH=1 END=getout ;
  INPUT paragraph $varying3200. 1;
  mailtext{paragraph_count} =paragraph;
  IF getout THEN LEAVE;
END;
* Read the addresses;
SET newsletter_admin END=eof ;
* Speed throttle. Tune messages per burst and wait time;
IF _n_/3 = INT(_n_/3) OR eof THEN DO; * Pause after every 3 messages;
  delay=SLEEP(5); * Wait 5 seconds;
END;
subject = 'Newsletter of '||PUT(TODAY(),MONYY5.);
FILE mail;
PUT '!EM_TO!' mailaddress;
PUT '!EM_SUBJECT!' subject;
PUT '!EM_FROM! info@synchrona.nl";
PUT '!EM_ATTACH!' "E:\SASForum\2013\newsletter.pdf";
* Following loop fills the body of the mail and substitutes the placeholders;
DO n=1 TO paragraph_count;
  Line= mailtext{n};
  DO WHILE (INDEX(Line,'####'));
    flag = INDEX(Line,'####');
    IF flag THEN DO;
      Code = SUBSTR(Line,flag+4,1);
      IF Code = '3' THEN DO;
        IF today() - max(Membersince,Renewaldate) GT 365 THEN
          line = substr(line,6);
        ELSE Line = '$$$$';
      END;
      IF CODE = '3' THEN CONTINUE;
      SELECT (Code);
        WHEN ('1') Substitute = Given_name;
        WHEN ('2') Substitute = Surname;
        WHEN ('3') ;
        WHEN ('4') Substitute = put(intck('month',
          max(Membersince,Renewaldate),today()),best12.);
        WHEN ('5') DO;
          if Renewaldate ne . then Substitute =
            'renewed your subscription';
          else Substitute = 'registered';
        END;
      OTHERWISE Substitute = '????';
    END;
    Line = COMPBL(SUBSTR(Line,1,flag-1)||
      Substitute||SUBSTR(Line,flag+5));
  END;
  END;
  IF NOT INDEX(line,'$$$') THEN PUT Line ;
END;
PUT '!EM_SEND!' / '!EM_NEWMSG!';
IF eof THEN PUT '!EM_ABORT!';
RUN;

```

Program 7: Mass-mailing a newsletter.

```

NOTE: The infile MAILMSG is:
      Filename=E:\SASforum\2013\mailtext.txt,
      RECFM=V,LRECL=256,File Size (bytes)=373,
      Last Modified=19Feb2013:20:50:19,
      Create Time=19Feb2013:15:37:28

NOTE: The file MAIL is:
      E-Mail Access Device

Message sent
      To:           'JSMITH@pops.com'
      Cc:
      Bcc:
      Subject:      Newsletter of FEB13
      Attachments:  "E:\SASForum\2013\newsletter.pdf"
Message sent
      To:           'PN@Northcorp.com'
      Cc:
      Bcc:
      Subject:      Newsletter of FEB13
      Attachments:  "E:\SASForum\2013\newsletter.pdf"
Message sent
      To:           'MarySunface@hotmail.com'
      Cc:
      Bcc:
      Subject:      Newsletter of FEB13
      Attachments:  "E:\SASForum\2013\newsletter.pdf"
Message sent
      To:           'amesbury.g@gmail.com'
      Cc:
      Bcc:
      Subject:      Newsletter of FEB13
      Attachments:  "E:\SASForum\2013\newsletter.pdf"
NOTE: 9 records were read from the infile MAILMSG.
      The minimum record length was 0.
      The maximum record length was 237.
NOTE: 57 records were written to the file MAIL.
      The minimum record length was 1.
      The maximum record length was 249.
NOTE: There were 4 observations read from the data set
      WORK.NEWSLETTER_ADMIN.
NOTE: DATA statement used (Total process time):
      real time           11.17 seconds
      cpu time            0.54 seconds

```

Listing 2: SAS Log messages from the mailing.

CONCLUSION

Sending mail from the SAS DATA Step is a simple but powerful feature, whether you want to send a single mail with or without an attachment or you want to send a personalized mail to all your friends or colleagues in the company.

In fact it would be easy to set up your own spamming operation. But this paper should not be seen as an encouragement to do so. On the contrary. Spamming is illegal and using smart software like SAS to do it does not change that.

REFERENCES

The FILENAME, FILE, PUT statements and options in these statements that are not handled here as they are rarely necessary, are documented in SAS® 9.3 Language Reference Dictionary. Some system-specific details can be found in SAS® 9.3 Companion for Windows.

The FOPEN and FINFO functions are also documented in SAS® 9.3 Language Reference Dictionary

A general description of the email interface can be found in SAS® 9.3 Language Reference Concepts.

The %SYSFUNC macro function is documented in SAS® 9.3 Macro Language Reference

The information about the OUTLOOK pop-up message can be found in the Microsoft Knowledge base under article ID: 2723336

For more information about using ODS, including options to customize the layout you are referred to the SAS® 9.3 Output Delivery System: User's Guide.

CONTACT INFORMATION

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

Name	Erik Tilanus
Address	Horstlaan 51
Postal code, City	3971 LB Driebergen
Country	the Netherlands
Work Phone:	+31 343 517007
Fax:	+31 842 152107
E-mail:	erik.tilanus@planet.nl
Web:	www.synchrona.nl

At the website you can also find other presentations by the author, held at previous SUGI and SAS Forum meetings.

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.