

## Paper 340-2013

## You have got SASMAIL! Simple SAS® Macro for sending e-mails.

Rajbir Chadha, Cognizant Technology Solutions

**ABSTRACT**

As SAS® software programs become complex, processing times increase. Sitting in front of the computer, waiting for the program to finish isn't something that everyone enjoys. This paper talks about a way to have SAS send out automatic e-mails once the process finishes and include the log, output or reports as an attachment. This function gives users the freedom to work on other pressing tasks rather than constantly checking the SAS process log for completion.

The SASMAIL custom macro function combines the functionality of FILENAME EMAIL, SAS Macros and the SQL Procedure to deliver the intended results. It sends out an e-mail with a copy of the log (when available) as an attachment providing users with the flexibility to access required information even when mobile.

SASMAIL detects the SAS session you are running, whether DMS or BATCH, and selects the attachment options accordingly. Although this has been designed for SAS running on UNIX, it can be easily tweaked to work on Windows® version of SAS as well. The SASMAIL macro uses a lookup for the user's login and e-mail address. This allows the macro to work with minimum or no arguments.

SASMAIL macro can be used by novice and advanced SAS users with little or no modification. The function allows users to customize what they want to see in the e-mail including the e-mail list, attachment and even the e-mail content. Users can even include a Microsoft® Excel® file or a summary report as the attachment.

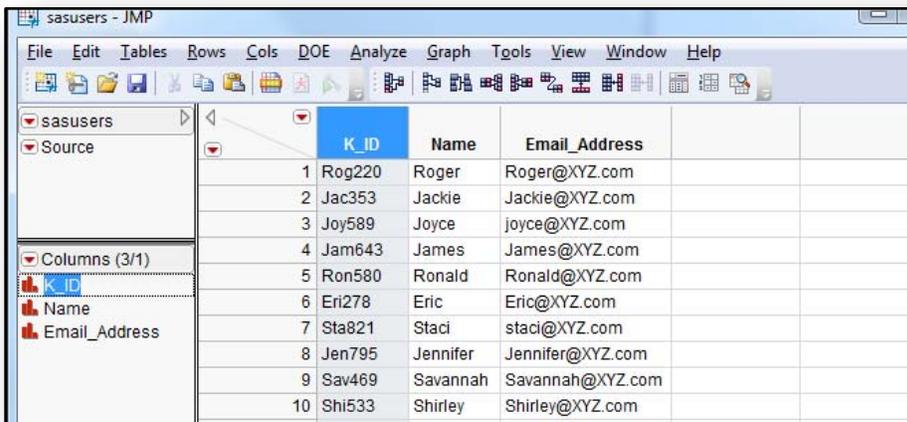
**INTRODUCTION**

This paper describes a custom macro function that triggers an e-mail when invoked. The macro function is driven by certain preset parameters which can be overridden by the users. This function can be used in batch programs as well as interactive SAS sessions, to e-mail the log file and/or output. Users don't have to sit and monitor long running programs and can view the log/output even from a mobile device (assuming they have access to their e-mail via mobile).

**BACKGROUND AND ASSUMPTIONS**

This paper assumes that users are familiar with the UNIX environment (i.e. file structure, environment variables and basic UNIX commands). The macro has been designed for a UNIX based SAS system, but with some minor modifications, it can work in other SAS environments as well.

This code requires a user list with login ids and corresponding e-mail addresses (**Figure 1**) to send e-mails when no parameters are specified.



	K_ID	Name	Email_Address
1	Rog220	Roger	Roger@XYZ.com
2	Jac353	Jackie	Jackie@XYZ.com
3	Joy589	Joyce	joyce@XYZ.com
4	Jam643	James	James@XYZ.com
5	Ron580	Ronald	Ronald@XYZ.com
6	Eri278	Eric	Eric@XYZ.com
7	Sta821	Staci	staci@XYZ.com
8	Jen795	Jennifer	Jennifer@XYZ.com
9	Sav469	Savannah	Savannah@XYZ.com
10	Shi533	Shirley	Shirley@XYZ.com

Figure 1. SAS data set for the login to e-mail lookup

## SASMAIL MACRO

You can send electronic mail programmatically from SAS using the FILENAME EMAIL access method. This macro simplifies the e-mail procedure and requires very less number of arguments to be supplied. The macro is included in the autocall library and is accessible to all users (Autocall macros will be explained later in this paper). This macro has various advantages:

- You can automatically send e-mail upon completion of a SAS program that you submitted for batch processing or interactive session. This is achieved simply by calling the macro without specifying any options.
- You can attach logs for the batch session or text based reports or outputs in the e-mail.
- Interactive session log can also be included but requires additional options to be specified (discussed later in this paper).
- Replies are sent to the 'from' e-mail and not to the UNIX account.

The SASMAIL macro uses 6 parameters to achieve the desired functionality. All the parameters are optional and may be omitted by users if the pre-requisites are met.

Parameter description and default values are described below.

**%sasmail(to="<e-mail list>", from="<e-mail list", subject="<text>", type="<mail type", attach="<file location>", mail\_text="<text>");**

1. **To:** is the e-mail address to which you want to send the message. Address must be enclosed in quotation marks. In order to specify more than one address, enclose each of them in quotation marks and separate them with a space. You may include the list of addresses in parenthesis. If you omit 'to', SAS extracts the e-mail address from the user list on UNIX server, based on your login.
2. **From:** specifies the e-mail address of the sender. Default value is the e-mail address of the user who is running SAS. Use this option when the person who is sending the message from SAS is not the author. You must enclose the address in quotation marks. You can specify only one e-mail address. To specify the author's name along with the address, enclose the address in angle brackets (< >). Here are some examples:

**from="jdoe@home.com"**

**from="John Doe <jdoe@home.com>"**

If you omit this parameter, SAS extracts the e-mail address from the user list on UNIX server, based on your login.

3. **Subject:** specifies the subject of the e-mail message. If the subject contains special characters or spaces, you must include the text in quotation marks. Here are some examples:

**subject=Sales**

**subject="June \$\$ Sales Report"**

If you omit subject, the following default subject message will be included, "*Mail from UNIX AIX -SAS account*".

4. **Type:** specifies the content type for the message body. If you do not specify any content type, SAS tries to determine the correct content type. You must enclose the value in quotation marks

**type='content/type'**

The default value of this parameter is 'text/plain'.

5. **Attach:** specifies the name of the file (or files) to be attached with the e-mail. It also specified the options to modify attachment specifications. The file name must be specified as a fully qualified path recognized by the operating system and should 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:

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

**attach="/sas/apps/sas\_prod/user/demo\_user/report.txt"**

```
attach=("/sas/apps/sas_prod/user/demo_user/June-July12.txt"
"/sas/apps/sas_prod/user/demo_user/July-August12.txt")
```

If you omit attach, the default attachment will be included. (SAS program log in case of batch session)

The attachment-options include the following:

- a. **CONTENT\_TYPE='content/type'**: Specifies content type for the attached file. You must enclose this 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.

'CT=' and 'TYPE=' may be used as aliases instead of 'CONTENT\_TYPE='.

If SAS cannot determine a content type based on the filename and extension, the default value is text/plain. For Microsoft Excel files CT="application/excel".

- b. **ENCODING='encoding-value'**: Specifies text encoding of the attachment that is read into SAS. You must enclose the value in quotation marks.
- c. **EXTENSION='extension'**: 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.

'EXT=' may be used as alias for 'EXTENSION='.

- d. **NAME='filename'**: 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")
```

- e. **OUTENCODING='encoding-value'**: Specifies the resulting text encoding for the attachment to be sent. You must enclose the value in quotation marks.

6. **mail\_text** specifies the text content of the mail. Text must be included in quotation marks.

If you omit mail\_text, SAS will include the default mail text message like, "Hi, this e-mail is being triggered by the UNIX-SAS process...".

## SASMAIL EXAMPLES

Here are some examples with SASMAIL using the various options available to the users.

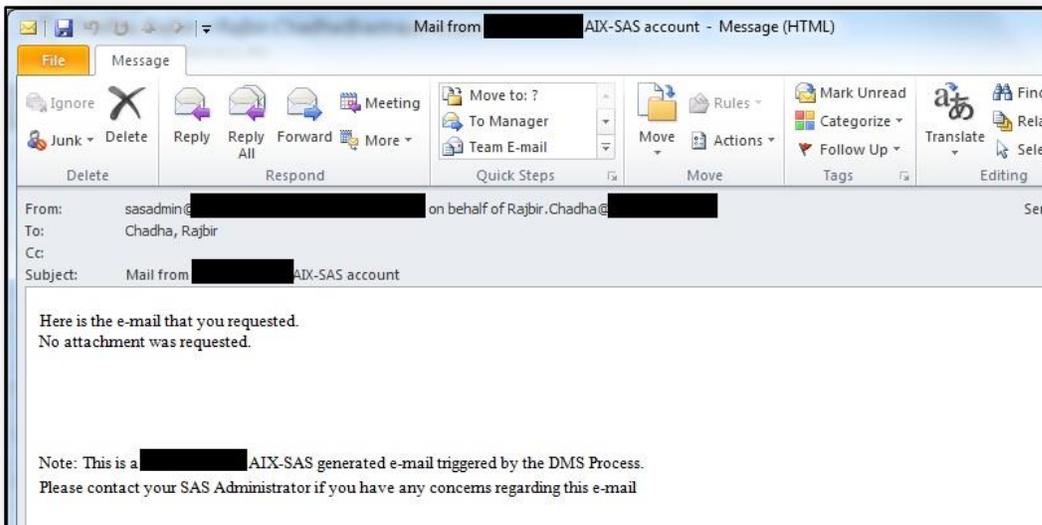
Example 1: Sending E-mail with default options

In order to get an automated trigger e-mail after the completion of your process, include the following macro at the end of the process. All default options will be used and an e-mail will be sent to the e-mail address mapped against your login id in the lookup data set.

```
%sasmail;
```

**Figure 2. SASMAIL Example - Sending E-mail with default options**

The e-mail is sent by the SAS server containing the default text, indicating the server name and the program/process name. The text in the screenshot below shows 'DMS Process' which denotes an Interactive SAS session. When the e-mail is triggered by a Batch SAS session, the program name is indicated in the e-mail. This information is pulled up from the SASHELP.VEXHFL table.



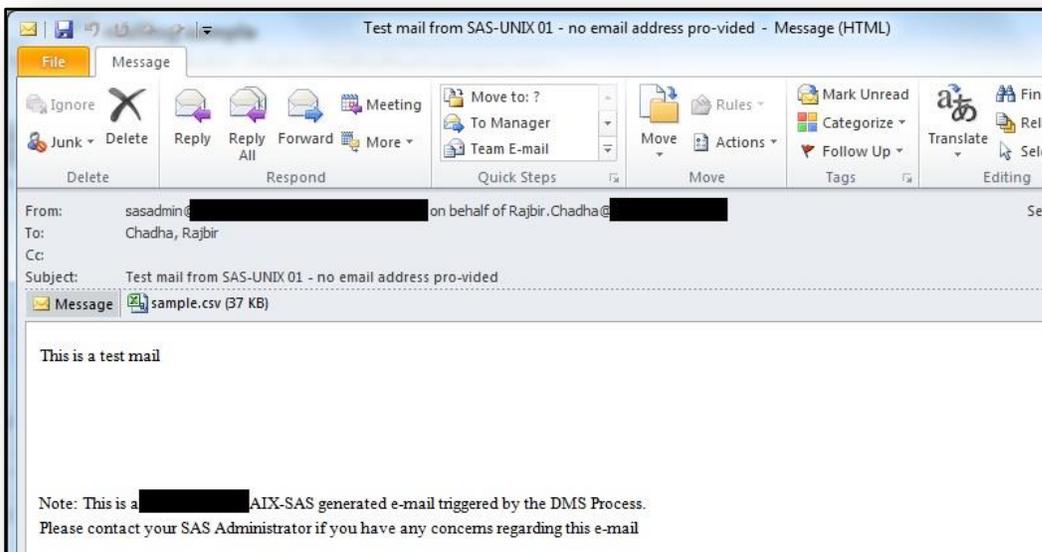
**Figure 3. E-mail with default options**

Example 2: Sending E-mail with attachment and no 'to' e-mail address

The following program sends a message and attachment to the user (based on the user id and e-mail mapping). For this example, the e-mail options specified are the 'subject=', 'attach=' and 'mail\_text='.

```
%sasmail (subject="Test mail from SAS-UNIX 01 - no e-mail address provided",
attach=("/sas/apps/sas_prod/user/demo_user/sample.csv"), mail_text="This is a test
mail")
```

**Figure 4. SASMAIL Example - Sending E-mail with attachment and no 'to' e-mail address**



**Figure 5. E-mail with attachment and no 'to' e-mail address**

### Example 3: Sending Procedure Output in E-mail

You can use e-mail to send the output of SAS procedures. This example shows how to send ODS HTML as an attachment with the e-mail message.

```

data temperatures;
  do centigrade = -40 to 100 by 10;
    fahrenheit = centigrade*9/5+32;
    output;
  end;
run;

ods html file="/sas/apps/sas_prod/user/demo_user/temp.html";

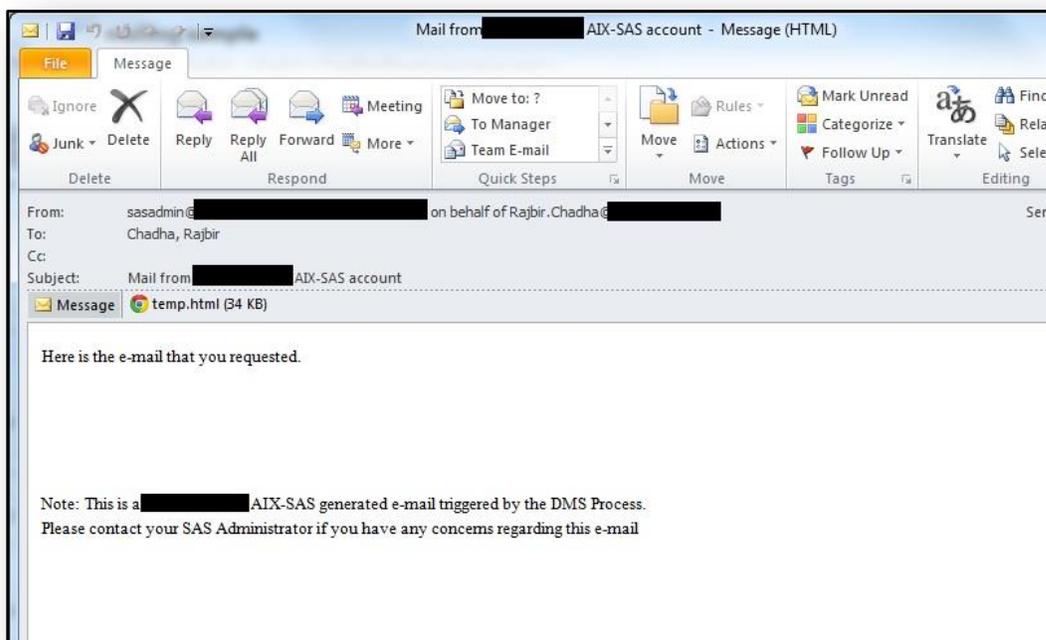
title 'Centigrade to Fahrenheit Conversion Table';

proc print;
  id centigrade;
  var fahrenheit;
run;
ods html close;

%sasmail(attach="/sas/apps/sas_prod/user/demo_user/temp.html");

```

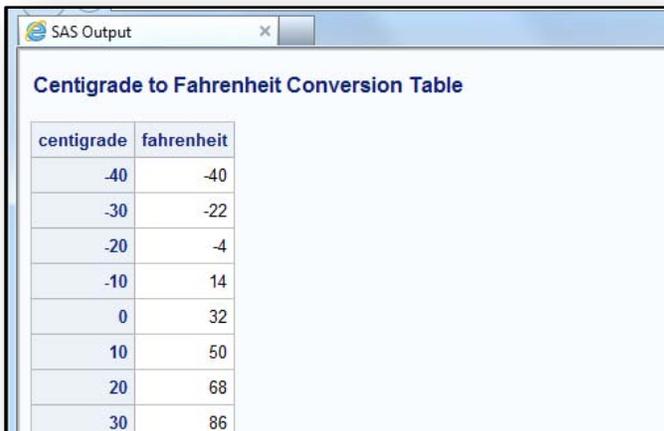
**Figure 6. SASMAIL Example - Sending Procedure Output in E-mail**



**Figure 7. E-mail with procedure output**

The HTML attachment included in the e-mail can be opened in your default browser (Internet Explorer or Google Chrome). You can add HTML formatting and options to the ODS statement to make the output look better. Images and graphs from other SAS procedures can also be directed to the HTML output and included as an attachment using the SASMAIL macro.

Below is the sample output from SAS using ODS HTML.



The screenshot shows a window titled "SAS Output" with a table titled "Centigrade to Fahrenheit Conversion Table". The table has two columns: "centigrade" and "fahrenheit". The data rows are as follows:

centigrade	fahrenheit
-40	-40
-30	-22
-20	-4
-10	14
0	32
10	50
20	68
30	86

**Figure 8. HTML output from Example3**

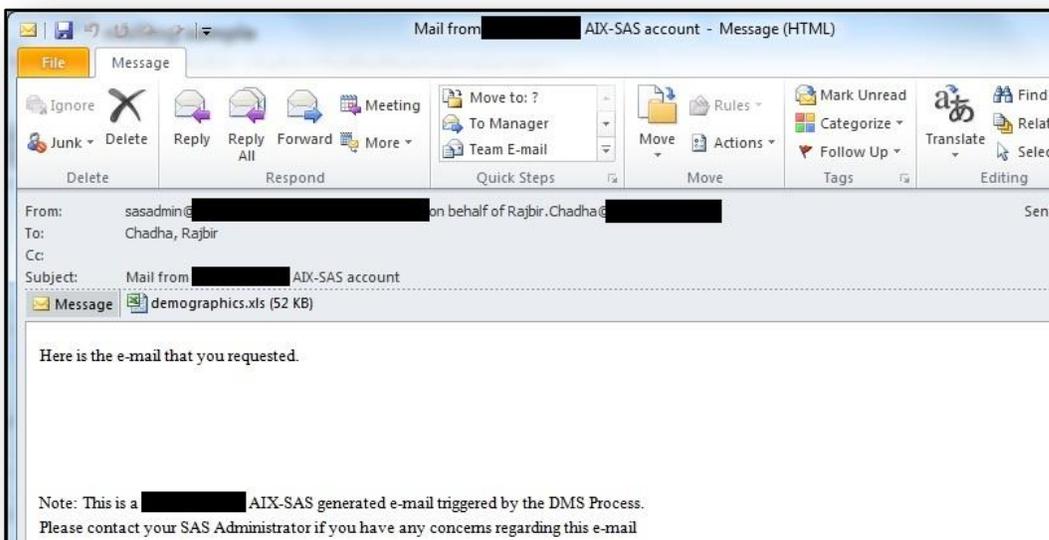
Example 4: Creating and E-mailing an excel file

The following example illustrates how to create a Microsoft Excel file and send it from SAS in an e-mail message. Note the use of Content Type (CT) option to classify the attachment.

```
proc export data=sashelp.demographics
  outfile="/sas/apps/sas_prod/user/demo_user/demographics.xls"
  dbms=xls replace;
run;

%sasmail (attach=("/sas/apps/sas_prod/user/demo_user/demographics.xls"
CT="application/excel"));
```

**Figure 9. SASMAIL Example - Creating and E-mailing a Microsoft Excel file**



**Figure 10. E-mail with Microsoft Excel attachment**

	A	B	C	D	E
1	CONT	ID	ISO	NAME	ISONAME
2		91	180	44 BAHAMAS	BAHAMAS
3		91	227	84 BELIZE	BELIZE
4		91	260	124 CANADA	CANADA
5		91	295	188 COSTA RICA	COSTA RICA
6		91	300	192 CUBA	CUBA
7		91	320	214 DOMINICAN REPUBLIC	DOMINICAN REPUBLIC
8		91	330	222 EL SALVADOR	EL SALVADOR
9		91	415	320 GUATEMALA	GUATEMALA
10		91	420	332 HAITI	HAITI
11		91	430	340 HONDURAS	HONDURAS
12		91	487	388 JAMAICA	JAMAICA
13		91	595	484 MEXICO	MEXICO
14		91	665	558 NICARAGUA	NICARAGUA
15		91	710	591 PANAMA	PANAMA
16		91	763	659 SAINT KITTS/NEVIS	SAINT KITTS AND NEVIS
17		91	926	840 UNITED STATES	UNITED STATES
18		92	149	28 ANTIGUA/BARBUDA	ANTIGUA AND BARBUDA
19		92	150	32 ARGENTINA	ARGENTINA

Figure 11. Microsoft Excel attachment from Example4

## INTERACTIVE-SESSION SAS LOG

When using interactive SAS session on UNIX, the log is displayed in the log window. This log is not stored on the UNIX server hence it cannot be e-mailed using the SASMAIL macro function. To work around this issue, you can use the 'ALTLOG' option while invoking the DMS SAS session. ALTLOG creates an alternate log file on the UNIX file system at the path specified in the argument. This is a permanent log for the Interactive SAS session.

Use the following syntax at the UNIX command prompt to launch a DMS SAS session with alternate log:

```
sas -altlog /sas/apps/sas_prod/user/demo_user/samplelog.log &
```

Use the following SASMAIL options to attach the alternate log in the e-mail.

```
%sasmail (attach="/sas/apps/sas_prod/user/demo_user/samplelog.log") ;
```

## SASMAIL CODE

The SASMAIL macro full code is shown below. The code uses NONOTES, NOSOURCE and NOSOURCE2 SAS system options to suppress the log entries. This prevents the macro and its processing steps to be printed in the log file. The code requires a lookup data set with e-mail address and corresponding user id. These values are used when no parameters are specified to the macro.

Table 1. SASMAIL Full Code

```

/*****
/* Program Name      : sasmail.sas
/* Purpose          : Read in the e-mail parameters and send out e-mail to the users*/
/* Owner/SME       :
/*
/* Change Log
/*****/

```

```

/* Comments      :                                          */
/* Use "-altlog" system option when launching interactive SAS to attach the log in*/
/* e-mail        :                                          */
/*****/

%macro sasmail(to, from, subject, type, attach, mail_text);
/* Options to suppress log entries */
options nonotes nosource nosource2;

/* Accessing SAS User list from core sasuser data set */
Libname core "/sas/adminscripts/data";
%let prid=&SYSUSERID.;

/* Extracting the e-mail address for the logged in user */
proc sql noprint;
    select email_address into :usermail
    from core.sasusers where k_id ="&prid.";
quit;
%put user_id is: %trim(&prid.) e-mail address is: %trim(&usermail.);

%if &to.= %then %let to="%trim(&usermail.)";
%if &from.= %then %let from="%trim(&usermail.)";
%if &subject.= %then %let subject="Mail from &syshostname. &SYSSCPL.-SAS account";

/* Debugging Statement - optional */
/* %put to: &to. from: &from. subject: &subject.; */
%let saspgm=;
%if &mail_text.= %then
    %let mailtext="Here is the e-mail that you requested.";
%else %let mailtext=&mail_text.;
%if &type.= %then
    %let type="text/plain";

/*Searching for log file to include in default attachment*/
%if %length(&attach.) le 0 %then %do;
proc sql noprint;
    select distinct xpath into :saspgm
    from sashelp.vextfl
    where upcase(xpath) contains ".LOG";
quit;
%let pgm=%trim(&saspgm.);
%if %length(&pgm.) le 0 %then %do;
    %let attach= ;
%end;
%else %do;
    %let attach="%trim(&pgm.)";
%end;
/*%put the attachment is &attach.; */
%end;

%if %length(&attach.) le 0 %then %do;
FILENAME MYFILE EMAIL (&to.)
        SUBJECT=&subject.
        TYPE=&type.
        emailid=&from.;

data _null_;
file myfile;
put &mailtext.;
put "No attachment was requested. " ;
put " " ;
put " " ;
put " " ;
put " " ;

```

```

    put "Note: This is a &syshostname. &SYSSCPL.-SAS generated e-mail triggered by the
&sysprocessname.." ;
    put "Please contact your SAS Administrator if you have any concerns regarding this
e-mail" ;
    run;
%end;

%else %do;
FILENAME MYFILE EMAIL (&to.)
        SUBJECT=&subject.
        TYPE=&type.
        emailid=&from.
        attach=&attach.;

data _null_;
file myfile;
put &mailtext.;
put " " ;
put "Note: This is a &syshostname. &SYSSCPL.-SAS generated e-mail triggered by the
&sysprocessname.." ;
put "Please contact your SAS Administrator if you have any concerns regarding this
e-mail" ;
    run;
%end;

/* Options to enable log printing */
options notes source source2;
%mend sasmail;

```

## STORED MACROS

The autocall macro facility stores the source code for SAS Macros in a single location on the file system, called autocall library. It is used for storing and maintaining macros in one place that can be easily accessed by users. Multiple autocall libraries can be combined together using SASAUTOS system option in SAS.

The first time that an autocall macro is called in a SAS session, macro is compiled by the macro processor. This may sometimes be considered as an overhead, but it can be overcome by using the stored compiled macro facility.

Stored compiled macro facility stores the already compiled macros in SAS catalogs in the specified SAS library. When using stored compiled macros, compilation is not required and macros are available as soon as the SAS session starts. Source codes for the stored compiled macros have to be maintained in a different location.

The autocall macro facility and the stored compiled macro facility each have their own advantages. You may choose one over the other based on some of the following factors:

- Frequency at which the macro will be used
- Frequency of changes to be made in the macro
- Number of users executing the macro
- Number of lines of code in the compiled macro

Programs that are under development should compile macros in the session. Production jobs that use stored compiled macros can be quite efficient. The autocall macro facility is useful when sharing macros between users.

## STORED COMPILED MACRO

Compiled macros can be stored in SAS in the SASMACR catalog. The default location for this catalog is WORK. Compiled macros are stored with an entry type of MACRO. Name of the MACRO entry is same as the macro name. SAS searches this catalog first when looking for a macro. Permanently stored compiled macros are saved in the SASMACR catalog but in a different physical location on the file-system.

SAS system options, MSTORED/NOMSTORED are used to control the ability to use the stored compiled macros. SASMSTORE option is used to specify the library used for the permanently stored macros. The following SAS statement turns on the stored compiled macro facility and assigns the library 'mymacr' for SASMSTORE.

```
options mstored sasmstore=mymacr;
```

Macros can be stored to the SASMSTORE location by using the option '/store' with the macro statement (when defining the macro).

```
%macro sample(var1, var2) /store;
```

## AUTOCALL MACRO FACILITY

The autocall library is searched after searching the SASMACR catalog. Autocall macro facility uses the SASAUTOS system option to specify any directory as an autocall library. Reserved fileref, SASAUTOS, is assigned at invocation time to the autocall library, designated by the site administrator. Multiple autocall libraries can be concatenated with the SAS default autocall libraries.

To create library members, store the source code for each macro in a separate file in a directory. The name of the file must be same as the macro name. For example, statements defining a macro '%demo' must be in a file named 'demo'.

SAS system options MAUTOSOURCE/NOMAUTOSOURCE are used to enable or disable the autocall macro facility.

```
options mautosource
sasautos=(mymacr promacr sasautos);
```

You can also use a FILENAME statement and the CATALOG argument to point to the catalog containing autocall macros. The code below assigns a fileref, MYMACRO, pointing to MYMACR.MYAUTOS catalog:

```
libname mymacr 'SAS-data-library';
filename mymacro catalog 'mymacr.myautos';
```

Macros are always searched in the following order:

1. WORK.SASMACR catalog
2. Stored Compiled macros
3. Autocall macros

## CONCLUSION

SASMAIL macro function simplifies the e-mail procedure in SAS by reducing the lines of code on the users' end. Users can add the macro function call in their code and get notified as the program completes. The ability to add attachments such as log, output or Microsoft Excel files with the e-mail gives the users easy and quick access to their results and analysis. The macro parameters can be easily changed to allow a better customization such as the list of recipients, subject, attachments, etc.

The macro can be stored by using the autocall macro facility or stored compiled macro facility and be made available to the users in the beginning of their SAS sessions.

## REFERENCES

- SAS Institute Inc. 2011. SAS® 9.2 Language Reference: Dictionary, Fourth Edition. (Pages referenced 1532-1541). Cary, NC: SAS Institute Inc. Reproduced with permission of SAS Institute Inc., Cary, NC.
- SAS Institute Inc. 2009. SAS® 9.2 Macro Language: Reference. (Pages referenced 111-116). Cary, NC: SAS Institute Inc. Reproduced with permission of SAS Institute Inc., Cary, NC.
- Carpenter, Arthur L. 2001. Proceedings of the 9th Western Users of SAS Software Conference (September, 2001). "Building and Using Macro Libraries". Available at <http://caloxy.com/papers/45-p17-27.pdf>
- Worden, Jeanina, and Philip Jones. 2004. Proceedings of the Twenty-Ninth Annual SAS® Users Group International Conference. Cary, NC: SAS Institute Inc. (May, 2004). "You've Got Mail – E-mailing Messages and Output Using SAS® EMAIL Engine". Available at <http://www2.sas.com/proceedings/sugi29/178-29.pdf>

## ACKNOWLEDGMENTS

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

I would also like to mention the following author whose papers have been referenced here - Arthur L. Carpenter, Jeanina Worden and Philip Jones.

## CONTACT INFORMATION

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

Rajbir Chadha  
Cognizant Technology Solutions  
Wilmington, DE 19810  
Phone: 1 (302) 290-6130  
E-mail: [Rajbir.Chadha@Cognizant.com](mailto:Rajbir.Chadha@Cognizant.com) / [Rajbir.Chadha@gmail.com](mailto:Rajbir.Chadha@gmail.com)

Microsoft, Excel and PowerPoint are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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.