



SENDING EMAILS IN SAS TO FACILITATE CLINICAL TRIAL

Frank Fan, Clinovo, Sunnyvale CA

WUSS 2011

Annual Conference October 2011

TABLE OF CONTENTS

1. ABSTRACT	3
2. INTRODUCTION.....	3
3. SYSTEM CONFIGURATION	3
4. SYNTAX TO SEND EMAILS VIA SAS.....	4
5. FIRST EXAMPLE, A SIMPLE EMAIL WITH NO ATTACHMENT	7
6. EMAIL WITH ATTACHMENT	10
7. MULTIPLE EMAILS WITH ATTACHMENT.....	12
8. SENDING EMAIL FROM A PROC STEP	14
9. CONCLUSION	18
10. REFERENCES	18

1. ABSTRACT

Email has drastically changed our ways of working and communicating. In clinical trial data management, delivering reports to users is part of our daily activities. Most of our colleagues prefer receiving reports by email rather than via the web or on a shared drive. Email is indeed one of the most efficient and convenient ways of communication.

In this paper, we will present how to use SAS V9 to send reports via email. Before an email can be sent via SAS, the SAS configuration file needs to be modified. After addressing the configuration file, SAS syntax to send an email will be discussed. We will use several examples to illustrate how we can implement such practices.

2. INTRODUCTION

As computer technology advances, more and more pharmaceutical companies are taking advantage of new technologies and computerizing the entire process of clinical data collection, data management, data analyses and FDA submission. Sending emails is one of those technologies so widely used that it is impossible to exclude it from any aspects of clinical trial.

We often send a lot of emails during the process of data management, including clinical data cleaning, monitoring, safety, and bio-statistics, and quite a bit of them are sent via SAS.

3. SYSTEM CONFIGURATION

To make use of the email function, the SAS configuration file needs to be modified. The configuration file usually resides in:

SAS-configuration-directory\Levn\SASApp\sasv9_usermods.cfg (Windows)

SAS-configuration-directory/Levn/SASApp/sasv9_usermods.cfg (UNIX and z/OS)

- EMAILSYS MAPI VIM SMTP Determines which application interface to use (MAPI or VIM), or indicates direct communication with the SMTP server (SMTP).
- EMAILID "logon id" May be requested by your email software when you specify MAPI or VIM under the EMAILSYS option.
- EMAILPW "password" May be requested by your email software when you specify MAPI or VIM under the EMAILSYS option.
- EMAILHOST "SMTP-server-domain-name" Specifies the exact domain name for your SMTP server. Ex. "mail.schardy.qc.ca"
- EMAILPORT port-number Specifies the physical port number used for SMTP communication. Generally, port 25 is used and it is the default value for this option.

Here is a sample configuration file:

```
-CONFIG "C:\Program Files\SAS\SAS 9.1\nls\en\SASV9.CFG"  
-emailsys SMTP  
-emailhost Avdmail.clinovo.com  
-emailport 25
```

4. SYNTAX TO SEND EMAILS VIA SAS

You can program SAS to send electronic mail from SAS using the EMAIL (SMTP) access method. To send email to an SMTP server, you first need to specify the SMTP email interface with the EMAILSYS system option, use the FILENAME statement to specify the EMAIL device type, and then submit SAS statements in a DATA step or in PROC step. This email access method has several advantages:

- You can use the logic of the DATA step or PROC step to subset email distribution based on a large dataset of email addresses.
- You can automatically send email upon completion of a SAS program that you submitted for batch processing.
- You can directly see the output through email based on the results of processing.

In general, a DATA step or PROC step that sends email has the following components:

- a FILENAME statement with the EMAIL device-type keyword
- email options specified in the FILENAME or FILE statement that indicate email recipients, subject, attached file or files, and so on
- PUT statements that define the body of the message
- PUT statements that specify email directives (of the form !EM_directive!) that override the email options (for example, TO=, CC=, SUBJECT=, ATTACH=) or perform actions such as send, end abnormally, or start a new message.

We have several ways to specify the email elements, such as subject, to, cc, attachment etc. We are going to describe them one by one.

`FILENAME fileref EMAIL <'address' ><email-options>`

This filename statement must be defined to send out emails via SAS. Device type keyword "EMAIL" specifies the EMAIL device type, which provides the access method that enables you to send electronic mail programmatically from SAS.

As an option, email addresses and other email options like attachment can be defined in the statement but can also be defined elsewhere. For more information regarding other options, see SAS Support 64316.

Another way to set the email addresses, to, cc, attachment information is to specify the email addresses and other email options when the fileref is used. If both places have specified addresses, the latter would overwrite the former.



Here is an example:

```
FILE outmail TO= "frank@clinovo.com" ATTACH="C:\\My SAS\\logsummary.html";
```

The email addresses and other email options can also be defined by using PUT statement to specify email directives that override the attributes of the message (the email options like TO=, CC=, SUBJECT=, CONTENT_TYPE=, ATTACH=). The syntax takes this format:

```
'!EM_TO! to-address'
```

This email directive 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! frank@clinovo.com';
```

```
put '!em_to! ("frank@clinovo.com " "stephen@clinovo.com");
```

```
put '!em_to! Frank Fan <frank@clinovo.com>;
```

Other email directives are listed below:

```
'!EM_BCC! bcc-address'
```

```
'!EM_CC! cc-address'
```

```
'!EM_FROM! from-address'
```

```
'!EM_IMPORTANCE! LOW | NORMAL | HIGH'
```

```
'!EM_REPLYTO! replyto-address'
```

```
'!EM_SUBJECT! subject'
```

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

The rules and usage are the same as “!em_to!”. For example,

```
put '!em_attach! /F/Protect/sout/Dashboard.xls';
```

```
put '!em_attach! ("F\\Protect\\sout\\Dashboard.xls" "F\\CHOICE\\sout\\Dashboard.xls")';
```



The following directives perform actions.

'!EM_SEND!' 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!' abnormally ends 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!' clears all attributes of the current message that were set using PUT statement directives.

5. FIRST EXAMPLE, A SIMPLE EMAIL WITH NO ATTACHMENT

In many cases, we are required to send an email notification when an event occurs. For example, we send out emails to related personnel for accountability purpose when a specific trial device is used. In case more than one device is used, the program will send out multiple emails. Hence a macro is used.

Although this task is usually realized in an EDC system, the EDC system may sometimes have issues and the release of bug fixes can take a long time for validation and implementation. With the help of Windows scheduled task, we automated this process. Below are the details.

First we generate a dataset as shown below. It includes the subject ID, device number, device reference code, form, site ID, type and a flag to indicate if it is a new entry or an old entry updated.

```
flagPATNUM    RSSTNTSERNOMSSTNTREFCODEformsiteidtype
NEW027530-100290416P1009FG1011943-020TL SI02753038
NEW027530-100290706P1002FG1011943-030TL SI02753038
```

Then we put all the information in several macro variables.

```
data _null_;
retain subj;
length subj $30;
set rpt end=eof;
if _n_=1 then subj=patnum;
else subj=trim(subj)||', '||trim(patnum);
call symput('flag'||compress(put(_N_,2.)), trim(flag));
call symput('form'||compress(put(_N_,2.)), trim(form));
call symput('patnum'||compress(put(_N_,2.)), trim(patnum));
call symput('siteid'||compress(put(_N_,2.)), trim(siteid));
call symput('sitename'||compress(put(_N_,2.)), trim(sitename));
call symput('type'||compress(put(_N_,2.)), 'Absolute Pro');
call symput('serno'||compress(put(_N_,2.)), trim(rsStntSerNo));
call symput('RefCode'||compress(put(_N_,2.)), trim(msStntRefCode));
if eof then do;
    call symput('row',_n_);
    call symput('subj',trim(subj));
end;
run;

%put &subj 'Row='&row;
```



In the following code, we define the filename with the keyword “EMAIL”, prepare the necessary email components, and send the emails out with adapted email content.

```
%Macro _output;
  %*let email_lst = "css@clinovo.com";
  %*let cc_lst = "css@clinovo.com" "frank.fan@clinovo.com";
  %let email_lst = "device.inventory@clinovo.com";
  %let cc_lst = "css@clinovo.com" "mobul@clinovo.com";
  %do i=1 %to &row;
filename mail EMAIL to=&email_lst
cc=(&cc_lst)
  from="css@clinovo.com" replyto="css@clinovo.com"
  subject="Device Usage Notification for MOBILITY - Patient &patnum&i "
  ;
data _null_;
  file mail;
  put " ";
  put "&flag&i "; put "----- ";
  put "The &form&i form indicates that subject &patnum&i at site
&siteid&i,
      &sitename&i, has reported use of the following device: &type&i.
";
  put " ";
  put "The Device Serial Number is: &serno&i. ";
  put " ";
  put "The Device Reference Code is: &refcode&i. ";
  put " ";
  put "Please check EDC for the most current information. ";
  put " ";
run;

%end;

%mend _output;

%_output;
```

The output email is displayed as shown in Figure 1.

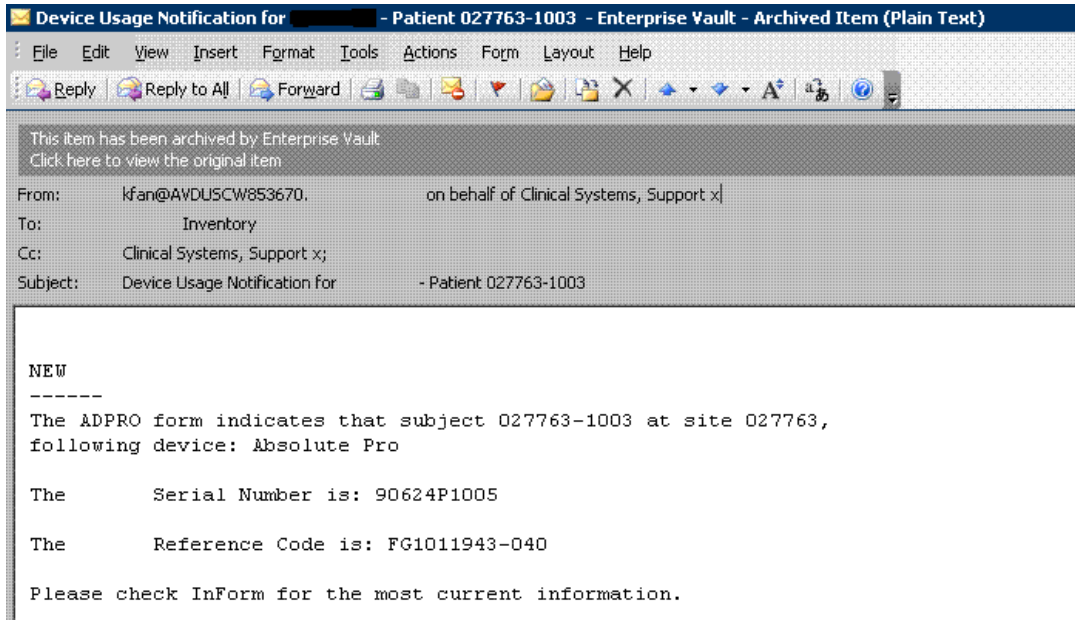


Figure 1. Sample email sent by SAS with no attachment

6. EMAIL WITH ATTACHMENT

In most of the cases, we need to send emails with attached reports to the recipients. The following is a typical email with report attached. This is an email we send weekly to users to report adverse events and adjudication results for reconciliation. Note that if the output file is empty, the email content will be different and there will be no attachment in the email.

In the program, macro variables &nob, &nob2 are the number of entries in 2 worksheets in the attachment Excel file. If both equals to zero, then the Excel file will not be attached to the email. If &nob is zero the email content will show as "STRIDES AE QC Check Report has been updated. No mismatch is found across all AE forms for all patients".

```
%macro _output;
%let email_lst="jennifer@clinovo.com" "marie@clinovo.com" "tatyana@clinovo.com"
"css@clinovo.com";
%*let email_lst= "css@clinovo.com" "Frank, Fan <frank.fan@clinovo.com>";
filename mymail email to=(&email_lst)
from="Stephen, Chan <Stephen.Chan@clinovo.com>"
replyto="Frank, Fan <kang.fan@clinovo.com>"
subject=" HECUES AE QC Check Report, &dt"
%if (&nobs gt 0) or (&nobs2 gt 0)
%then attach("&outpath\STRIDES AEQC Check &dt..xls" );
;
data _null_;
file mymail;
put 'Hi all,';
put " ";
%if &nobs gt 0 %then put "STRIDES AE QC Check Report has been updated.
Attached is a copy of the latest HECUES AE QC Check Report. ";
%else put "STRIDES AE QC Check Report has been updated. No mismatch is found
across all AE forms for all patients";;
put " ";
put "Generated from HECUES AEQC system on &dtm"; /* &dtm is the datetime
like Jul 18, 2011 ( 2:30) */
run;
%mend _output;
%_output;
```

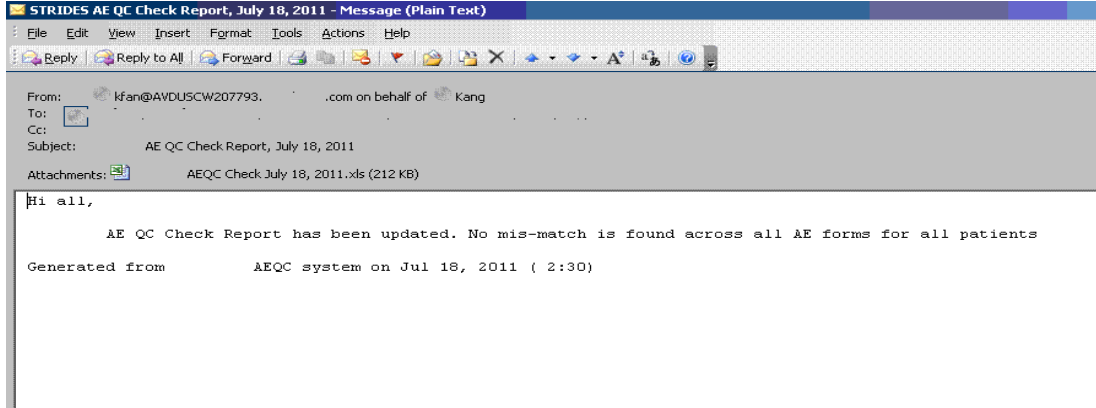


Figure 2. Sample email send by SAS with attachments

7. MULTIPLE EMAILS WITH ATTACHMENT

Here is another example of sending multiple emails with a macro. In conducting clinical trials, follow-up is a very important activity. Late or missed follow-up visits may cause compliance issues. To speed up the process and minimize errors, we developed a program that automates this process, by sending site email notification with a site-specific attachment to remind of subject follow-up visits. It includes the visit status, with overdue, in-window, in-coming visits. We first generate datasets for the information about overdue, in-window and in-coming visits, and then call the following macro to send the emails.

```
%macro GenerateSiteEmail(site_number);

data _null_;
  %nobs(QueryOverDue&site_number,OverDue_count);
  %nobs(QueryDue&site_number,Due_count);
  %nobs(Queryupcoming&site_number,upco_count);
run;

%IF (&Due_count ne 0 or &OverDue_count ne 0 or &upco_count ne 0) %THEN %DO;
  data doemail;
  set etemplate(where=(siteid=&site_number));
  by siteid;
  attrib pre_subject length=$100;
  attrib count length=8;
  length rtext $1000;

  *** Place values into macro variables ***;
  call symput('site_id',trim(siteid));
  call symput('rc_firstname',trim(rc_fn));
  call symput('site_name',trim(site_name));
  call symput('rc_lastname',trim(rc_ln));
  call symput('rc_email',trim(rc_email));
  call symput('rc2_email',trim(rc1_email));
  call symput('rc3_email',trim(rc2_email));
  call symput('cra1_firstname',trim(cra_name));
  call symput('cra1_lastname',);
  call symput('cra1_email',trim(cra_email));
  call symput('field_cra_email',trim(field_cra_email));
  call symput('DM_email',trim(DM_email));
  call symput('cra1_phone',trim(cra_phone));
  call symput('cra1_fax',trim(cra_fax));
  call symput('cra1_title',trim(cra_title));
  call symput('report_date',left(put(input("&sysdate",date9.),worddate20.)));

  if RC1_fn='' then address="Dear "|| trim(RC_fn)||",";
  else if RC2_fn='' then address="Dear "|| trim(RC_fn)||" and " ||
    trim(RC1_fn)||",";
  else address="Dear "|| trim(RC_fn)||", " || trim(RC1_fn) || " and " ||
    trim(RC2_fn) || ",";

  call symput('address',trim(address));

  *** Retain some values ***;
  retain e_to e_cc e_subject e_reply e_from;
  if row_code_column = 'to' then e_to = trim(resolve(text_column));
  if row_code_column = 'cc' then e_cc = trim(resolve(text_column));
  if row_code_column = 'reply_to' then do;
  e_reply = trim(resolve(text_column));
  e_from = trim(symget("cra1_firstname"))||" "||'<'||trim(e_reply)||'>';
```



```
end;

filename _email email LRECL=1000 RECFM=V
        attach="&siterep\ACT1_fup_compliance&&site_&i._&sysdate..pdf";

file _email;

if row_code_column = "subject" then do;
e_subject = resolve(trim(text_column));
    end;

if "&mode"="production" then do;
if compress(e_to) ne '' then
put '!EM_TO! ' e_to;
if compress(e_cc) ne '' then
    put '!EM_CC! ' e_cc;
end;
else do;
if compress(e_to) ne '' then
put '!EM_TO! ' 'css@clinovo.com';
if compress(e_cc) ne '' then
put '!EM_CC! ' "css@ clinovo.com";
end;

if compress(e_from) ne '' then
    put '!EM_FROM! ' e_from;

if compress(e_reply) ne '' then
    put '!EM_REPLYTO! ' e_reply;

put '!EM_SUBJECT! ' e_subject;

if row_code_column = "body" then do;
    rtext = resolve(text_column);
    if index(rtext,"- 0 ") eq 0 then put rtext;
end;

if last.siteid then do;
    put '!EM_SEND!';
    put '!EM_NEWMSG!';
    put '!EM_ABORT!';
end;
    run;
%END;
%mend GenerateSiteEmail;
%let mode=test;
%macro GSitesEmails(daytosend);
%IF (%SYSFUNC(day(%STR('%)&sysdate%STR('%)d)) eq &daytosend) or
    (%SYSFUNC(lowercase(&SYSDAY)) eq %SYSFUNC(lowercase(&daytosend))) %THEN %DO;
%do i=1 %to &nsite;
%GenerateSiteEmail(&&site_&i);
%end;

%END;
%mend GSitesEmails;
%GSitesEmails(17);
```



8. SENDING EMAIL FROM A PROC STEP

With the option EMAILSYS=SMTP, it is possible to send out emails from PROC step instead of DATA step.

Here is an example. Currently, we have almost 200 programs scheduled to run automatically. To maintain the system and monitor the status, we developed a program that scans the SAS program logs and emails us the result daily.

The email part is as follows:

```
OPTIONS LINESIZE=256;

FILENAME output EMAIL SUBJECT= "%sysfunc(date(),yymmdd10.) log summary -
%TRIM(&nOK) of %LEFT(&nmember) passed" FROM= "Clinical Systems Support
<css@clinovo.com>"

TO= "css@av.abbott.com" CC=("eva.li@clinovo.com" "Kang.Fan@clinovo.com"
"stephen.chan@clinovo.com") CT= "text/html";* Required for HTML output;

ODS HTML BODY=output STYLE=sasweb;
TITLE JUSTIFY=left

"%EVAL(&nmember-&nOK) of your program(s) are generating errors. &morningrun
programs ran this morning.";

PROC REPORT DATA=logsummary NOWD
STYLE (REPORT)=[PREHTML="<hr>"];

COLUMNS Study Program Results Last_run Size;*Inserts a rule between title &
body;

RUN;

ODS HTML CLOSE;
```

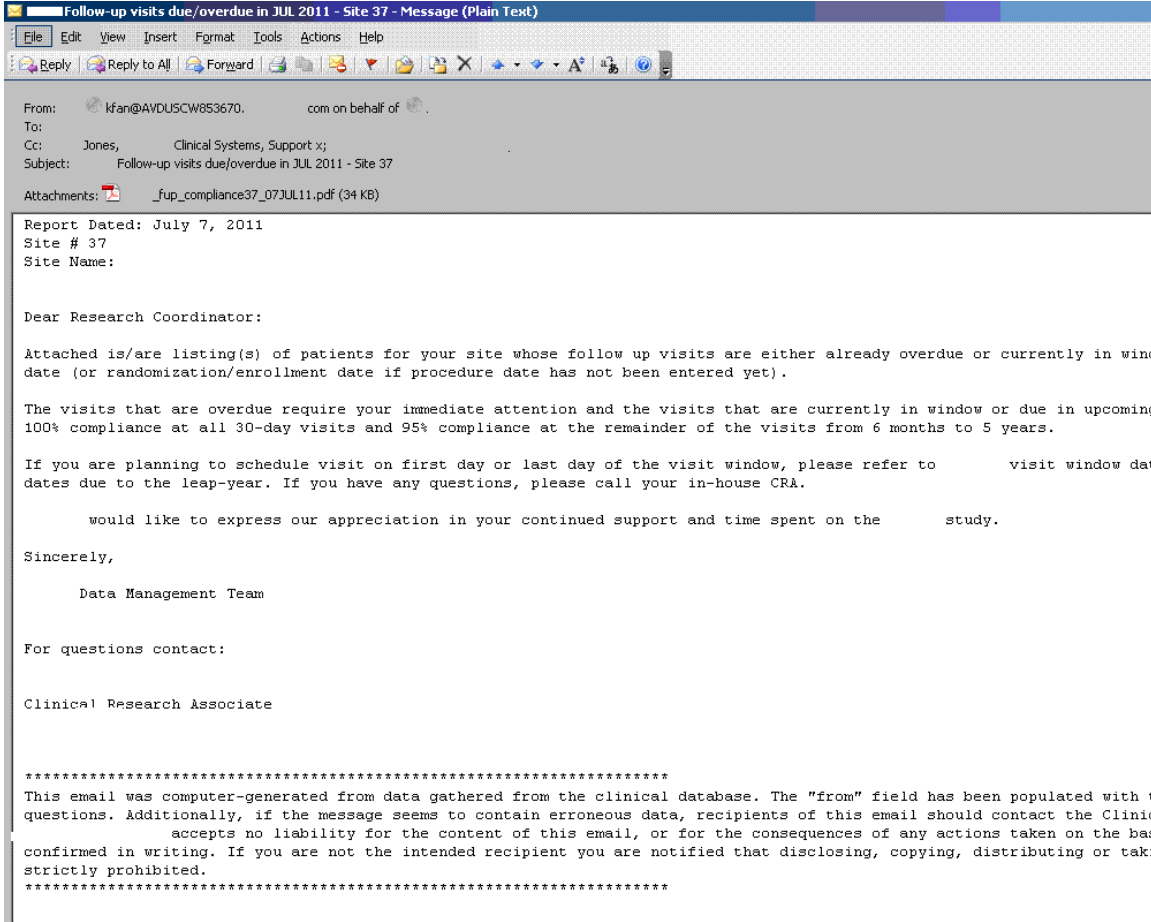


Figure 3. Sample emails send by SAS with attachments

From: kfan@AVDUSCW853670.clinovo.com; on behalf of Clinical Systems, Support x
 To: Clinical Systems, Support x
 Cc: Li, Eva; Fan, Frank N; Chan, Stephen
 Subject: 2011-07-08 log summary - 185 of 189 passed
 4 of your program(s) are generating errors. 42 programs ran this morning.

study	Program	Results	Last_run	Size
General	DashboardEmail	The Log is Clean of Any Errors	08JUL11:08:35:00 AM	34,402

study	Program	Results	Last_run	Size
General	ScheduleTasks	The Log is Clean of Any Errors	06JUL11:05:00:00 AM	282,914
General	Trial1DashboardEmail	The Log is Clean of Any Errors	05JUL11:06:15:00 AM	33,668
General	UpdateGraphs	The Log is Clean of Any Errors	04JUL11:06:40:00 AM	816,169
General	extract_viper	The Log is Clean of Any Errors	01JUL11:10:33:00 AM	6,678
General	importMedDat2011AR	The Log is Clean of Any Errors	01JUL11:10:47:00 AM	223,265
CRUST	pt_profiles_dataset	The Log is Clean of Any Errors	08JUL11:07:25:00 AM	108,832
CRUST	dset_mover	The Log is Clean of Any Errors	27OCT09:10:25:00 AM	64,503
CRUST	pt_profiles_dataset.sas	The Log is Clean of Any Errors	11MAR09:09:13:00 AM	88,706
HECUES	hecues_profiles_dataset	The Log is Clean of Any Errors	08JUL11:07:31:00 AM	159,775
HECUES	QCreport	The Log is Clean of Any Errors	08JUL11:04:04:00 AM	226,528
HECUES	editchecks	The Log is Clean of Any Errors	08JUL11:03:59:00 AM	5,693,584
HECUES	Crfstatus	The Log is Clean of Any Errors	08JUL11:03:55:00 AM	3,153,918
HECUES	hecues_query_report	The Log is Clean of Any Errors	08JUL11:03:40:00 AM	70,109

study	Program	Results	Last_run	Size
HECUES	MEds_listings	The Log is Clean of Any Errors	08JUL11:03:16:00 AM	82,482
HECUES	DashBoardNew	The Log is Clean of Any Errors	08JUL11:03:05:00 AM	3,374,380
HECUES	HECUES_dset_trim	The Log is Clean of Any Errors	08JUL11:12:33:00 PM	1,411,967
HECUES	Import	The Log is Clean of Any Errors	08JUL11:12:29:00 PM	417,642

Figure 4. A sample email send by SAS Proc step



9. CONCLUSION

Sending email from SAS is convenient and useful to clinical trial conduction. It is possible to send one or multiple emails, with or without attachments. As described, if you follow the right syntax and have the system configured properly, it is flexible to send emails for different purposes, and you can speed-up and automate your clinical trials

10. REFERENCES

Further information about the FILENAME, FILE, PUT and the email interface can be found in SAS 9.1 Language Reference: Concepts and SAS 9.1 Language Reference: Dictionary

CONTACT INFORMATION

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

Name: Frank Fan
Enterprise: Clinovo
Address: 1208 East Arques Avenue
City, State ZIP: Sunnyvale CA 94085
Work Phone: 408-416-1636
Email: frank.fan@clinovo.com
Web: www.clinovo.com

At the website you can also find other presentations and papers regarding clinical trials. 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.