

Paper 039-29

Automated distribution of SAS® results

Jacques Pagé, Les Services Conseils HARDY, Quebec, Qc

ABSTRACT

This paper highlights the programmable aspects of SAS® results distribution using electronic mail facilities. It provides a clear understanding of all options, statements and commands for that purpose. Also provided are means for accessing data from your Outlook® or Notes® address books.

INTRODUCTION

Now that SAS (ODS) can produce publication-ready results in various formats, it is also quite convenient to use SAS in order to distribute those results via email.

It is then possible to entirely automate the production and distribution of SAS results.

SUPPORTED STANDARDS AND MAIL ENGINES

- ▶ MAPI (Messaging Application Program Interface)
Used by Microsoft products (Outlook)
- ▶ VIM (Vendor Independent Mail)
Used by IBM products (Notes)
- ▶ SMTP (Simple Mail Transport Protocol)
Provides a direct communication between SAS and your SMTP server, i.e. without passing through a mail software.

These are actually considered as "SAS mail engines".

CONTROL

SAS email facilities are controlled by email options or commands at four different scope levels:

1. System options
2. FILENAME statement options
3. FILE statement options (DATA step)
4. PUT statement directive commands (DATA step)

MAIL RELATED SYSTEM OPTIONS

GENERAL

- ▶ **-EMAILSYS** *MAPI* *VIM* *SMTP*
Determines which application interface to use (MAPI or VIM), or indicates direct communication with the SMTP server (SMTP). See figure 1 below.
- ▶ **-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.

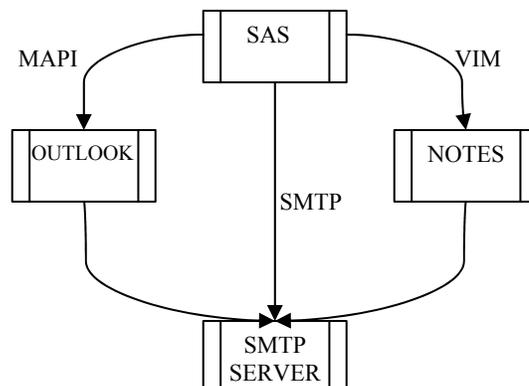
SMTP SPECIFIC

- ▶ **-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.

WHERE TO SPECIFY?

- ▶ All those system options are **configuration** or **start-up** options.
- ▶ They must be specified in the SAS configuration file (SasV8.cfg) or in the SAS start-up command.
- ▶ It is therefore impossible to specify those options on a regular **OPTION** statement.

FIGURE 1 – MAIL ENGINES USAGE



FILENAME STATEMENT

- ▶ A **FILENAME** statement is required for sending email from a DATA or PROC step.
- ▶ Must precede the step that actually sends the email.
- ▶ Must specify the **EMAIL** access method.

FILENAME STATEMENT SYNTAX

FILENAME *fileref* **EMAIL** <"*address*"> <*options*> ;

- *Fileref* is a name by which to which DATA or Proc steps will refer to in order to send email.
- *Address* is the recipient's email address (can also be specified as an option).
- *Options* will provide normal email specifications

FILENAME STATEMENT OPTIONS

- ▶ **CC= "address"**
CC= ("address-1" "address-n")
email address of someone to receive a copy of the email
- ▶ **FROM= "address"** author's address
- ▶ **SUBJECT= "subject-of-the-message"**
- ▶ **ATTACH= "access path and name of an attached file"**
ATTACH=("access path and name of attached file 1" "access path and name of attached file n")
- ▶ **REPLYTO= "address"**
REPLYTO= ("address" "address")
- ▶ **TO= "address"**
TO= ("address" "address")
recipient's email address(es)
- ▶ **CT= "text/plain" or "text/html"**
(with EMAILSYS=SMTP only) for Version 8
The "text/html" value of this option indicates that the body of the message is HTML formatted but does not actually format it into HTML code.
- ▶ **EMAILID= "logon-id"**
- ▶ **EMAILPW= "password"**
- ▶ **EMAILSYS= MAPI | VIM | SMTP**
These last three options are normally specified as system options from the SAS configuration file. The FILENAME statement lets us override them.

FILENAME STATEMENT EXAMPLES

```
FILENAME outmail EMAIL charlie@sas.com
subject="Results as of 2003-07-14";
```

```
FILENAME outmail EMAIL charlie@sas.com
subject="Results as of 2003-07-14"
attach="C:\My Folder\pers.html";
```

EMAILING FROM THE DATA STEP

The DATA step uses the **FILE** statement to direct output to a mail destination specified on a previous **FILENAME** statement. It also uses **PUT** statements to actually write the contents of the message, i.e. the message body.

GENERIC EXAMPLE

```
FILENAME outmail EMAIL
  SUBJECT="Results as of %sysfunc(date(),yymmdd10.)"
  FROM=   "bob@sas.com"
  TO=     "charlie@sas.com"
  ATTACH= "C:\My Folder\test.html";

DATA _NULL_;
  FILE outmail;
  PUT "Here are the results for your last query";
RUN;
```

INTEGRATED EXAMPLE

```
/* Identifying the results file */
%LET results=C:\My Folder\test.html;

/* Producing the actual results (HTML format) */
ODS HTML BODY="&results" STYLE=sasweb;
TITLE1 "Average of salaries by department and function";
PROC TABULATE DATA=mylib.pers;
  CLASS dept function;
  VAR salary;
  TABLE dept,function*mean*salary;
RUN;
ODS HTML CLOSE;

/* Global email options */
FILENAME output EMAIL
  SUBJECT="Results as of %sysfunc(date(),yymmdd10.)"
  FROM=   "bob@sas.com"
  TO=     "charlie@sas.com"
  ATTACH= "&results";

/* Sending email */
DATA _NULL_;
  FILE output;
  PUT "Here are the results of your last query (attached HTML file)";
RUN;
```

FILE STATEMENT FOR EMAILING FROM THE DATA STEP

A **FILE** statement is required for emailing from the DATA step. It directs output from the **PUT** statement(s) to a mail destination specified on a previous **FILENAME** statement.

FILE STATEMENT SYNTAX

FILE *fileref* <*mail options*>;

- *fileref* must correspond to a *fileref* defined on a previous **FILENAME** statement
- *mail options* are optional. If used, they override corresponding mail options specified at the **FILENAME** level.

FILE STATEMENT EXAMPLE

```
/* Common mail options */
FILENAME outmail EMAIL
  SUBJECT="Results as of %sysfunc(date(), yymmdd10.)"
  FROM=   "bob@sas.com";

DATA _NULL_;
  FILE outmail
    TO=    "charlie@sas.com"
    ATTACH="C:\My Folder\test1.html";
  PUT "Here are the results of your query";
RUN;

DATA _NULL_;
  FILE outmail
    TO=    "alice@sas.com"
    ATTACH="C:\\My Folder\test2.html";
  PUT "Here are the results of your query";
RUN;
```

In this example, two DATA steps are used to send different mail messages (the attached file being different) to different persons.

EMAILING FROM A PROC STEP (HTML OUTPUT)

Through ODS, the results of any PROC step may be sent as the body of a mail message without having to use a DATA step. Currently, only HTML may be used as the ODS destination and this requires using the **EMAILSYS=SMTP** option.

EXAMPLE

```

/* Global email options */
FILENAME output EMAIL
  SUBJECT= "Sales summary report 1"
  FROM=    "bob@sas.com"
  TO=      "charlie@sas.com"
  CT=      "text/html" /* Required for HTML output */ ;

ODS HTML BODY=output STYLE=sasweb;
TITLE JUSTIFY=left
  "Total Sales by Country and Product type as of %sysfunc(date(),yymmdd10.)";

PROC REPORT DATA=sashelp.prdsale NOWD
  STYLE(REPORT)=[PREHTML="<hr>"] /*Inserts a rule between title & body*/;
  COLUMNS COUNTRY PRODTYPE,ACTUAL;
  DEFINE COUNTRY /GROUP;
  DEFINE PRODTYPE /ACROSS;
  DEFINE ACTUAL/SUM "";
  BREAK AFTER PRODTYPE /SUMMARIZE;
RUN;
ODS HTML CLOSE;

```

FIGURE 2 - HTML MESSAGE AS SEEN UNDER OUTLOOK


Total Sales by Country and Product type as of 2003-12-30

	Product type	
	FURNITURE	OFFICE
Country		
CANADA	\$97,864.00	\$149,126.00
GERMANY	\$101,194.00	\$144,804.00
U.S.A.	\$91,567.00	\$145,782.00

DIRECTIVE COMMANDS FROM THE PUT STATEMENT

Within a DATA step, the **PUT** statement accepts **directive commands** that can override some mail options specified at the **FILENAME** or **FILE** level.

That facility allows dynamic modification of mail options for each iteration of the DATA step. It is therefore possible, for example, to modify the recipient's address (*TO= option*) according to the current value of a recipients dataset – in other words implementing a mail merge application.

DIRECTIVE COMMANDS SYNTAX

```
PUT directive-command <value>;
```

COMMON DIRECTIVE COMMANDS

These commands will override the corresponding mail options specified at the FILENAME or FILE level:

- ▶ **!EM_ATTACH!**
- ▶ **!EM_CC!**
- ▶ **!EM_FROM!**
- ▶ **!EM_REPLYTO!**
- ▶ **!EM_SUBJECT!**
- ▶ **!EM_REPLYTO!**

SPECIAL DIRECTIVE COMMANDS

These commands control the actual sending of the message and the persistence of the dynamic values.

- ▶ **!EM_SEND!**
- ▶ **!EM_ABORT!**
- ▶ **!EM_NEWMSG!**

When sending email thru a DATA step, an implicit **!EM_SEND!** command is executed at the end of each iteration – very similar to the implicit OUTPUT statement for writing the current observation into the output dataset.

If the actual sending of the message has to be conditional, then an explicit **!EM_SEND!** command must be executed. Contrary to the OUTPUT statement analogy however, the implicit **!EM_SEND!** command will still be executed; the **!EM_ABORT!** command must then be specified at the bottom of the DATA step in order to prevent implicit sending .

The **!EM_NEWMSG!** command is used for clearing any dynamic mail option specified in the current cycle.

MAIL MERGE EXAMPLE

```
FILENAME outbox EMAIL FROM="bob@www.sas.com"
  SUBJECT="Analytical results as of December 12, 2003"
  ATTACH="C:\Statistics\stat20031212.html";

DATA _NULL_;
  FILE outbox;
  /* Read recipients data */
  SET contacts (KEEP= sex name email_add);
  PUT "!EM_TO!" add_email;
  IF sex = "F" THEN PUT "Mrs " name ",";
  ELSE IF sex = "M" THEN PUT "Mr " name ",";
  PUT " ";
  PUT "Here are the periodic analytical results for 2003-12-12";
  PUT " ";
  PUT "The results are contained in an HTML file that can be";
  PUT "opened under Excel";
RUN;
```

ACCESSING YOUR OUTLOOK® OR NOTES® CONTACT FILES

With MAPI or VIM engines, you can use any symbolic recipient name or even group names defined in the contact files. It is however impossible to get access to other fields of recipients data (sex, title, geographic address, etc.).

When SMTP engine is used, no direct access to your contact files is provided.

This section will show how Outlook® or Notes® address books may be accessed from your DATA step.

READING THE OUTLOOK® CONTACT FILES

- ▶ No standard ODBC interface (surprisingly!)
- ▶ Possibility of manually exporting to a delimited file, but that task would have to be repeated in order to get fresh data.
- ▶ The following Vbs script extracts some meaningful data from the default contact file and exports them into a delimited file. The script is executable from a SAS program through an X statement.

```
'This script creates a CSV file containing one record for every
'contact within the default Outlook contact file

'Starting Outlook
Set outl = CreateObject("Outlook.Application")
'Getting access to Outlook folders
Set MyNameSpace = outl.GetNameSpace("MAPI")
'Obtaining the default contact file
Set Cont = MyNameSpace.GetDefaultFolder(10)
'Creating the delimited file
Set fso = CreateObject("Scripting.FileSystemObject")
'Opening the file
Set f = fso.OpenTextFile("c:\My Folder\contacts.csv", 2, True)
For Each Item In Cont.Items
  'Writing a record for each contact
  f.writeline Item.FullName & "," _
    & Item.Title & "," _
    & Item.EmailAddress
Next
```

The following program is an example of a SAS program invoking the previous script:

```
/* Invoking the VBS script */
OPTION NOXWAIT;
X "C:\My scripts\outlook_csv.vbs" ;
/* Converting the CSV file produced by the script into a SAS dataset */
DATA contacts;
  INFILE "C:\My Folder\contacts.csv" DLM="," DSD;
  LENGTH contact $ 40
         title $ 3
         email_add $ 100;
  INPUT contact title email_add;
RUN;
FILENAME outbox EMAIL FROM="bob@sas.com"
                SUBJECT="Analytical results as of December 12, 2003"
                ATTACH="C:\Statistics\stat20030912.html";

DATA _NULL_;
  FILE outbox;
  /* Reading contacts datasets */
  SET contacts (KEEP= title email_add);
  PUT "!EM_TO!" email_add;
  PUT title ",";
  PUT " ";
  PUT " Here are the results of your last query (attached HTML file)";
  PUT " ";
  PUT "These results are contained in an HTML file that can be opened";
  PUT "under Excel";
RUN;
```

READING THE NOTES® ADDRESS BOOK

- ▶ The NOTESSQL ODBC driver, supplied by Lotus, provides access to Lotus Notes® databases, including the address book database.

- ▶ The driver must be installed on your machine and an ODBC data source must be configured.

- ▶ DATA may be accessed through a LIBNAME statement or using SQL Pass-through.

The following program uses SQL Pass-through in order to access a Notes address book.

The ODBC data source, named "*notes_book*" in this example, must have been configured on your machine.

```
PROC SQL;
  CONNECT TO ODBC ("DSN=notes_book");
  CREATE TABLE contacts AS
  SELECT * FROM CONNECTION TO ODBC
    (SELECT * FROM address_book);
  DISCONNECT FROM ODBC;
RUN;
```

CONCLUSION

Distributing SAS results thru email is ...

- ▶ Easy
- ▶ Cost effective
- ▶ Reliable
- ▶ Highly programmable

May All Implement Likewise!

CONTACT INFORMATION

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

Jacques Pagé

Les Services Conseils HARDY
4715 des Replats, bureau 260
Québec, Qc Canada G2J 1B8
Work Phone: (418) 626-1666
Fax: (418) 626-2097
Email: jpage@schardy.qc.ca
Web: www.schardy.qc.ca

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.