Paper 084-2010

Zip and Email Files Using SAS® To Reduce Errors and Make Documentation Easy

Ryan Whitworth, RTI, International, Durham, NC

ABSTRACT

This paper describes an approach for using SAS® to create zip files and e-mail them as attachments to recipients from a contact list. The advantage of this approach is that it minimizes the potential for human error from manually zipping and e-mailing files, makes documentation simple, and reduces the labor burden. This approach was accomplished using Base SAS® Version 9.1 tools, free WinZip software, and an SMTP server.

INTRODUCTION

Each quarter, a set of approximately 2,500+ reports for approximately 24 recipients are created as rtf files. These reports must be placed into separate zip file packages that are then e-mailed to the appropriate grantee contact person. The average grantee receives a zipped file with a set of 100 reports along with a single overall report for that grantee.

Since the final report deliverables are output into a standard directory structure using a consistent file naming convention, we knew a SAS® macro solution might eliminate the need to manually zip and e-mail report packages to each grantee. We also recognized that using SAS® would allow us to keep a log file to better document the work that was done to zip and e-mail these reports.

Basically, the inputs we started with at the beginning of our approach are the sets of 2,000+ RTF report files broken down into groupings by grantee and a single overall report for each grantee. We needed to complete two different steps with our approach:

1. The files needed to be zipped into appropriate groupings (i.e. by grantee).
2. These zip files needed to be e-mailed to the appropriate recipient(s).

Fortunately, past SUGI papers addressed each of these steps separately so it was basically a matter of putting the two concepts together within one macro.

ZIPPING FILES IN SAS®

At the time that our approach was developed, there were two different ways of zipping files using SAS® that we identified in the existing SAS® literature. The first approach was presented at SAS® Global Forum 2007 (Paper 005-2007), and it uses command-line invocation of the WinZip32 executable. This approach was ultimately used to solve our problem using this basic syntax:

```
x "c:\progra~1\winzip\winzip32.exe [-min] action [options] filename[.zip] files"
```

or as will be used later in the example syntax:

```
x " ""c:\program files\winzip\WINZIP32.EXE"" -a
""\\server\directory\Deliverable.zip"" ""\\server\directory\*.rtf"" ";
```

Note that macro variables can be used within the command-line syntax and that wildcards (e.g. "*.rtf") are allowed. These points are important to our solution. Please read the Global Forum paper for more details regarding this process.

The other approach to creating a zip file in SAS® that we identified uses the ODS package syntax new to SAS® 9.2. This might have provided a more elegant solution, but we will explain why it did not work for our purpose at the end of this paper.

E-MAILING FILES IN SAS®

A SUGI 29 poster paper outlined the steps needed for our second step (Paper 170-29). We used the solution

presented in this paper to customize our final macro syntax that delivers the zip files to the appropriate recipients. Much like the proposed future work mentioned near the end of that paper, we developed an e-mail contact spreadsheet that can be easily updated and maintained by project staff and then imported into SAS®. Once the contact information is read into a temporary SAS® data set, we use the information within our final macro to link the zipped deliverable to the appropriate recipient.

The following is a generalized version of the piece of code used to create an e-mail file with an attachment using SAS®:

```
options EMAILSYS=SMTP EMAILHOST=example.smtpserver.org;

FILENAME MsgTxt EMAIL
TO=("recipient@sample.com")
CC=("archive@sample.org")
SUBJECT="TEST"
ATTACH=("\\server\directory\Deliverable.zip");

DATA _NULL_;
FILE MsgTxt;
PUT @1
//"Hello.";
run;
```

As noted in the SUGI 29 poster paper mentioned above, this code will open MS Outlook on the machine where it is run and may cause that software to generate alert messages warning that another program is attempting to create and send an Outlook e-mail message. A user can manually "OK" this action after each e-mail is generated, or an SMTP server can be used to avoid this problem. Please read the poster paper for more details.

COMBINING THE TWO CONCEPTS

The following code is a simplified version of our final macro. This version will create and attach only a single zip file, but a few simple additions made it possible to meet the needs of the deliverable reports described in the introduction:

```
options noxwait EMAILSYS=SMTP EMAILHOST=example.smtpserver.org;

*Import the contact data from an Excel spreadsheet;
PROC IMPORT OUT= WORK.contacts
            DATAFILE= "\\server\directory\ContactList.xls"
            DBMS=EXCEL REPLACE; RANGE="Sheet1$"; GETNAMES=YES; MIXED=NO;
            SCANTEXT=YES; USEDATE=YES; SCANTIME=YES;
RUN;

*Populate some macro variables using the contact data;
DATA _NULL_; SET contacts END=EOF; BY grant_number;
    CALL SYMPUT('grant_number'||LEFT(PUT(_N_,8.0)),TRIM(grant_number));
    CALL SYMPUT('address'||LEFT(PUT(_N_,8.0)),TRIM(email));
    IF EOF THEN CALL SYMPUT('max_loop',_N_);
RUN;

%macro ZipAndEmail;
%local i grant_dir reports_spec reports_zip;
*DO LOOP to run through all contacts from above;
%DO i=1 %TO &max_loop.;
%let grant_dir = C:\\Reports\Output\&rptdate.\&&grant_number&i..;
%let reports_spec = &grant_dir.\*.rtf;
%let reports_zip = &grant_dir.\&&grant_number&i.._reports.zip;
x " ""c:\program files\winzip\WINZIP32.EXE"" -a ""&reports_zip.""
""&reports_spec."" ";
FILENAME MsgTxt EMAIL
TO=("&&address&i..") CC=("AButMe@test.org")
SUBJECT="Reports for &&grant_number&i.." ATTACH=("&reports_zip.");
DATA _NULL_; FILE MsgTxt;
PUT @1 "DO NOT REPLY TO THIS MESSAGE."
```

```
//"The attached zip file contains reports for &&grant_number&i.."
//"If you have any problems, please contact: AnyoneButMe@rti.org";
run;
%END;
%mend ZipAndEmail;
%ZipAndEmail;
```

As you can see, the first portion of the code sets various options in SAS® include the noxwait option. This option specifies that the command processor returns to the SAS® session after the command is executed without having to type EXIT to close Winzip (in our example). The other options specifies that an SMTP server is to be used and gives its name and location.

Second, the contact information spreadsheet is imported. This spreadsheet contains important variables such as the grant number (organization/recipient) and email address ("address" in this example). It could also import many other fields from the contact spreadsheet. For example you could import the recipient's name to insert within the e-mail text to help personalize the message.

Next, user-defined global variables are established so they can be called in subsequent code. The data null step that follows populates several of those important global variables based on the contact information data set. This step also sets the "max_loop" variable that will serve to note the end of the DO loop that follows.

Perhaps the most important part of the code creates and executes a macro with a DO loop. This loop cycles through each record (e-mail recipient) from the contact list spreadsheet. At the top of this loop, a few additional macro variables are populated. They establish the location of the Winzip executable file, the name of the zip file to create, and the location of the reports that correspond to the current cycle/recipient. Note that the WinZip executable file must be in the location specified in this code on the machine where this code is being run. Also note that a wildcard character (*) is used to tell SAS® that we want to zip all of the ".rtf" files within a specific directory.

After a recipient's reports have been zipped, the FILENAME command creates and sends an e-mail. Other macro variables are used within this statement to customize the e-mail text and attachments for the recipient before advancing to the next cycle of the loop.

SPECIAL CONSIDERATIONS

Documentation requirements will vary for different types of jobs that can use this code, but with the methods described above it could be as simple as saving a SAS® log file. Admittedly, there is little more than documentation that the code was run for the zip portion of this process but this could be an improvement compared to manually running a zip program and leaving no documentation of the process. This email portion of the process is documented slightly more detail. An example showing documentation of a successful send is shown below:

```
NOTE: The file MSGTXT is:
    E-Mail Access Device
Message sent
      To:          "recipient@hotmail.com"
      Cc:
      Bcc:         "archive@rti.org"
      Subject:     TEST
      Attachments: ( "\\server\directory\Deliverable.zip" )
NOTE: 1 record was written to the file MSGTXT.
      The minimum record length was 0.
      The maximum record length was 221.
NOTE: DATA statement used (Total process time):
      real time           10.46 seconds
      cpu time            4.40 seconds
```

There are many other aspects of the logistics of e-mail delivery that must be considered. Handling delivery failures and reply messages from recipients are just a few. Finally, error messages in the log file must be carefully managed to identify any unexpected results (e.g. importing unexpected value formats or characters from the contact spreadsheet).

CONCLUSION- A PROMISING, POTENTIAL ALTERNATIVE (ODS PACKAGE)

The ODS package statement gives users another option for creating zip files within SAS®, and it does not require WinZip software. Therefore, the code could be batch submitted on any machine as long as it had SAS® Version 9,2 or higher (the ODS package statement available in SAS® 9.2). However, the logistics of our reporting process led us to choose to go outside of SAS® and use VBA to produce the final reports that were the input for our process described above. If we chose to keep all of our code within SAS®, then we could've used the ODS package statement for much more efficient delivery. For more information about the ODS package statement, please reference SAS® Global Forum 2009 Paper 018-2009.

REFERENCES

- Hunt, Stephen and Fairfield-Carter, Brian. "Zipping Right Along: Push-button SAS® Transfers via Command-Line Invocation of the WinZip32 Executable" *SAS® Global Forum 2007*. April 2007.
- Clanton, Sharon and Davis, Darlene. "Throw Away the Paper Trail and Send it by E-Mail" *Proceedings of the Twenty-Ninth Annual SAS® Users Group International Conference*. May 2004.
- Gebhart, Eric. "ODS Packages: Putting Some Zip in Your ODS Output" *SAS® Global Forum 2009*. March 2009..

ACKNOWLEDGMENTS

The author would like to acknowledge the help and support from his colleagues particularly Annette Green, Christine Davies, Joey Morris, and Robert Rabb.

CONTACT INFORMATION

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

Name: Ryan Whitworth
Enterprise: RTI, International
Address: 3040 East Cornwallis Road
City, State ZIP: Durham, NC 27709
Work Phone: 919-541-6086
E-mail: rwhitworth@rti.org