

## The Variety of Email Notifications Using SAS

Emily Morin, Blue Cross Blue Shield of Rhode Island

### ABSTRACT

Whether updating data or creating reports, an email notification is needed. Doing this manually has the opportunity for errors on multiple fronts, especially if multiple emails need to be sent out; further, writing multiple email notifications can take time away from other tasks. To keep a consistent email message, SAS® Enterprise Guide and SAS® Base programming can be used to simplify and keep consistency throughout messages. This paper will demonstrate the variety of ways to create email notifications that are easy to maintain. This paper is intended for the SAS® novice who is looking to simplify the process of repeated and consistent notifications for internal and external use.

### INTRODUCTION

Many workers may identify with the administrative burden of reports due in an on-going basis with a set cadence. With the completion of these tasks, an email is usually sent to alert stakeholders regarding readiness. While Outlook can save email templates, SAS® can make sending these emails even easier. After knowing the basics of creating an email, there are several tricks to make an email message simply to update users when needed. Building on the earlier work on this subject (Tilanus, 2008), there are several important coding differences that are used today. This paper explains how to create a simple email, add an attachment, and embed an image into an email and macros.

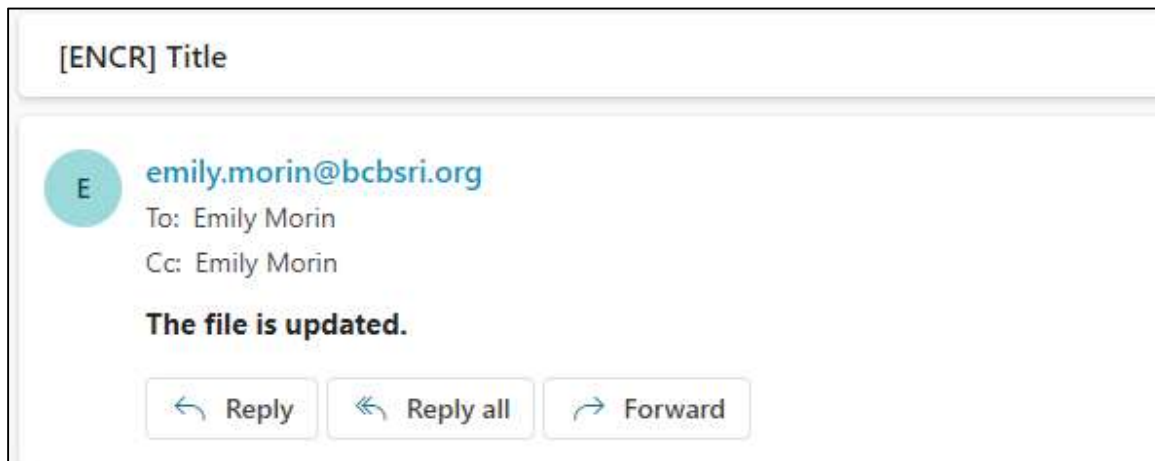
### TOPIC 1: CREATING A SIMPLE EMAIL

Before getting into tips, it is helpful to review how to create a basic email. First, start with *filename send email* to indicate creating an email. From there, use *from=* (") and *sender=* (") where then sender email is placed. Only one email can be used. Whether the coder is the sender or another coworker, this email should also be put in the quotes in the cc section along with the coder. If this is not done, the sender will not see this inbox when the code is executed.

The *to=* (") *statement* is where recipients of the email should be put. Each recipient should be surrounded by double quotes and a space between each email. Type indicates that the email is text based. The subject line has the title. If the email contains sensitive information *[encr]* should be put in the double quotes to encrypt the email. The code below shows the basic email:

```
filename sendmail email
  to= ("emily.morin@bcbsri.org")
  cc= ("emily.morin@bcbsri.org")
  from= ("emily.morin@bcbsri.org")
  sender= ("emily.morin@bcbsri.org")
  type="text/html"
  subject=" [ENCR] Title ";

data _null_;
  file sendmail;
  put '<html>';
  put '<body>';
  put '<b>';
  put 'The File is updated.';
  put '</body>';
  put '</html>';
run;
filename sendmail clear;
```



**Figure 1. Screenshot of result of email code, Basic email**

To break lines into different paragraphs, the put `<br>`; statement can be used. Without this command, each sentence in the file sendmail statement will stay as one paragraph. To put a sentence on a different line, only one `<br>` needs to be added.



**Figure 2: Screenshot of email message without break, `<br>`**



**Figure 3: Screenshot of email message result when one `<br>` is added**

With "put `'<br><br>';`" there is a one space in between the two lines it is between. One can add more than two `<br>` if a large space between them sentences is needed. The code placed in between the sentences in the email body code.

```
put '<br><br>';
```



Figure 4: Screenshot of email message result when two <br> is added

## TOPIC 2: ADDING AN ATTACHMENT

Sending documents and reports to others is an important skill. Within the email code, the line of code *attach* is where a file can be added:

```
attach=("path\file" content_type = "content_type")
```

The attach code goes in between the sender line and type lines:

```
filename sendmail email
to= ("emily.morin@bcbsri.org")
cc= ("emily.morin@bcbsri.org")
from= ("emily.morin@bcbsri.org")
sender= ("emily.morin@bcbsri.org")

attach= ( "\\path\file" lrecl=9999 content_type="content_type")
type="text/html"
subject="[ENCR] Title ";
```

For example, if attaching an Excel file the attach statement would be written as shown below:

```
attach=("Z:/fruit/oranges/2021/orange_may2021.xlsx" content_type="application/xlsx")
```

The path to the file allows SAS® to find the file needed to attach to the email. The '*content\_type*' statement allows SAS® to distinguish the type of file sent. Without this piece of code, the file will run into issues sending, such as error messages.. Below is a list of the possible types of attachments that can be applied to the *content\_type* statement.

| Extension | Content_type             |
|-----------|--------------------------|
| bmp       | image/bmp                |
| csv       | application/vnd.ms-excel |
| doc       | application/msword       |
| exe       | application/octet-stream |
| gif       | image/gif                |
| htm       | application/html         |
| html      | applicaton/html          |
| jpeg      | image/jpeg               |
| jpg       | image/jpeg               |
| log       | text/plain               |
| pdf       | application/pdf          |

|          |                               |
|----------|-------------------------------|
| png      | image/png                     |
| ppt      | application/vnd.ms-powerpoint |
| sas7bdat | application/sas               |
| tar      | application/x-tar             |
| text     | text/plain                    |
| txt      | text/plain                    |
| xls      | application/excel             |
| xlsx     | application/xlsx              |
| zip      | application/x-zip-compressed  |

**Table 1: Extension content type table from the article ‘Send Emails With Attachments Via SAS’**

In addition to attaching a file, images can be imbedded into emails. To do this, the attach statement is used, with *inlined* replacing content\_type:

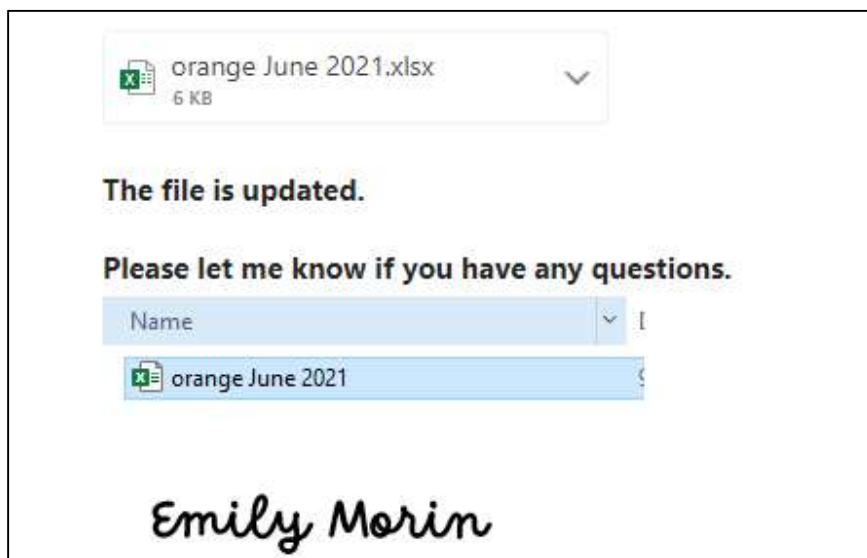
```
attach=("(path\file" inlined='name')
```

In addition, in the email message portion needs a *put statement* with the name put in the *inlined statement* in the body section of the email:

```
put '<img src=cid:name>';
```

This can include graphic files in PNG, GIF, JPEG and SAS/GRAPH® procedures (such as GCHART).

Multiple images can be imbedded in an email message. An image can be imbedded at the same time as attached files.



**Figure 5: Screenshot of email with multiple images imbedded and one file attached**

### TOPIC 3: ZIPPING MULTIPLE / TOO BIG FILES

With data, files can be big enough where it cannot be sent in an email. Adding multiple files can also be overwhelming for the email recipient to view. The solution to both issues is taking advantage of zip files. Through zipping one large or multiple files, one compressed file can be created. This process shrinks the file size and keeps the email code simpler. The code below shows the basic format of creating a zip file:

```
filename newfile "file_path/Zip_name.zip";
data _null_;
  if (fexist('newfile')) then
    rc = fdelete('newfile');
run;
filename newfile clear;
options dlcreatedir;
libname out "file_path/data";
ods package(newzip) open nopf;
ods package(newzip) add file="\\file_path\file&type";
ods package(newzip) add file="\\file_path\file1&type1";
ods package(newzip) add file="\\file_path\file2&type2";

ods package(newzip) publish archive properties
  (archive_path="\\file_path\final" archive_name="file_zip_name.zip");

ods package(newzip) close;
```

## TOPIC 4: USING MACROS

There are many projects sent multiple times throughout a year. Oftentimes, the message and files rarely change. Applying macros to the email code helps maintain this process. Macros are added at the beginning of the email code. This allows for one spot to update code instead of going through the entire code each time and keeps modifying code to a minimum. The result would be identical to Display 5. The code below shows how to apply macros:

```
%let release_date = June 2021;
%let create = June;

... attach=( "\\path\file&create." lrecl=9999 content_type="content_type")

type="text/html"
subject="[ENCR] Title &release_date. ";

data _null_;
  file sendmail;
  put '<html>';
  put '<body>';
  put '<b>';
  put 'The File for &release_date..';
  put '</body>';
  put '</html>';
run;
filename sendmail clear;
```

## CONCLUSION

Email is a useful tool to notify others internal and external to an organization regarding on-going reports. Setting up email notifications through SAS® can reduce the communication time. By investing in the setup and testing of the process upfront, efficiency may be gained and administrative burden lessened. It is important for the coder to have only their email address in the to, sender, cc, and from to test the code before sending the final copy. Adding macros and zipped files may take time to set up and test initially. This investment in time into macros and zipped file can lead to quicker time updating in future runs.

## REFERENCES

1. Bhalla, Deepanshu. "Send Emails With Attachments Via SAS." Accessed September 10, 2023. <https://www.listendata.com/2016/04/send-emails-with-attachments-via-sas.html>

2. Mincey, Martin. "Use SAS to send an email that embeds a graph in the body of the email." Accessed September 2, 2023. <https://blogs.sas.com/content/sgf/2017/04/21/use-sas-send-email-embeds-graph-body-email/>
3. Tilanus, Erik W. 2008. "Sending E-mail from the DATA step". *Proceedings of the SAS Global Forum 2008*, <https://support.sas.com/resources/papers/proceedings/pdfs/sgf2008/038-2008.pdf>.

## ACKNOWLEDGMENTS

The author extends her thanks to the SESUG 2023 Conference for accepting my abstract, paper, and being selected as a Professional Development grant recipient. The author extend thanks to manager, Karen Wallace for guidance and assistance.

## RECOMMENDED READING

- *Base SAS® Procedures Guide*
- *SAS® For Dummies®*

## CONTACT INFORMATION

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

Emily Morin  
Blue Cross & Blue Shield of Rhode Island  
[emily.morin@bcbsri.org](mailto:emily.morin@bcbsri.org)

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.