Clean Distribution and Emails: A Reporting Dream

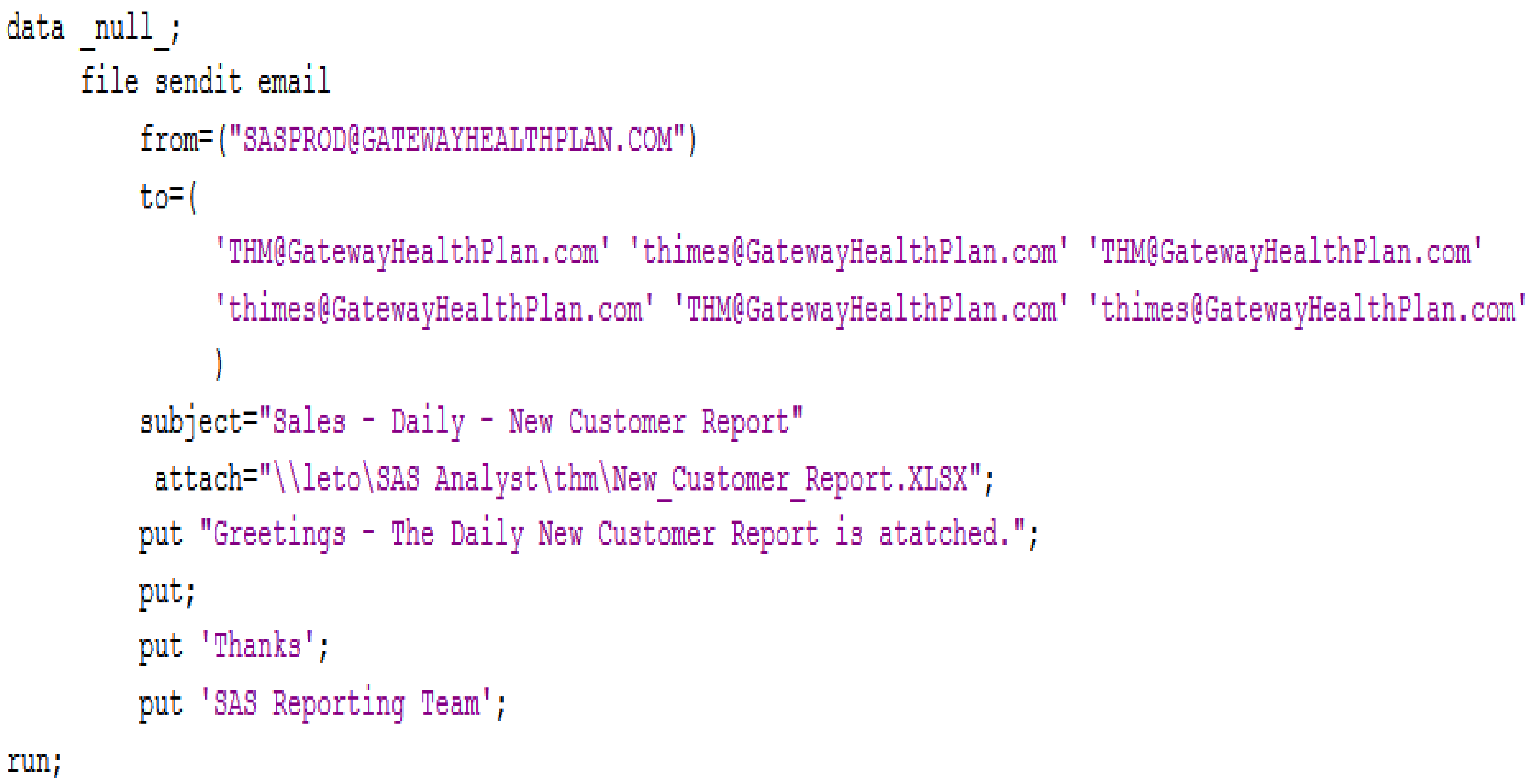
Ryan Bixler (rbixler@gatewayhealthplan.com) and Travis Himes (thimes@gatewayhealthplan.com)

Gateway Health Plan

1. Abstract

In a perfect world of reporting, all code is automated and successfully invoked without manual intervention or ever touching the SAS production code. Unfortunately, this is not the case for all reports across a company. This paper demonstrates how to efficiently add/remove recipients to a SAS report distribution list without manipulating the SAS production code. To do so my coworker and I developed an email distribution list along with a macro to send SAS reports. Together, these two macros provide a solution to efficient reporting and also create a report inventory. Using these macros allows users to get away from altering SAS production code to simply add/remove users who receive a report which is a best practice for most companies. We plan to educate users on how to develop and use these macros efficiently.

2. Standard Email for Production Code

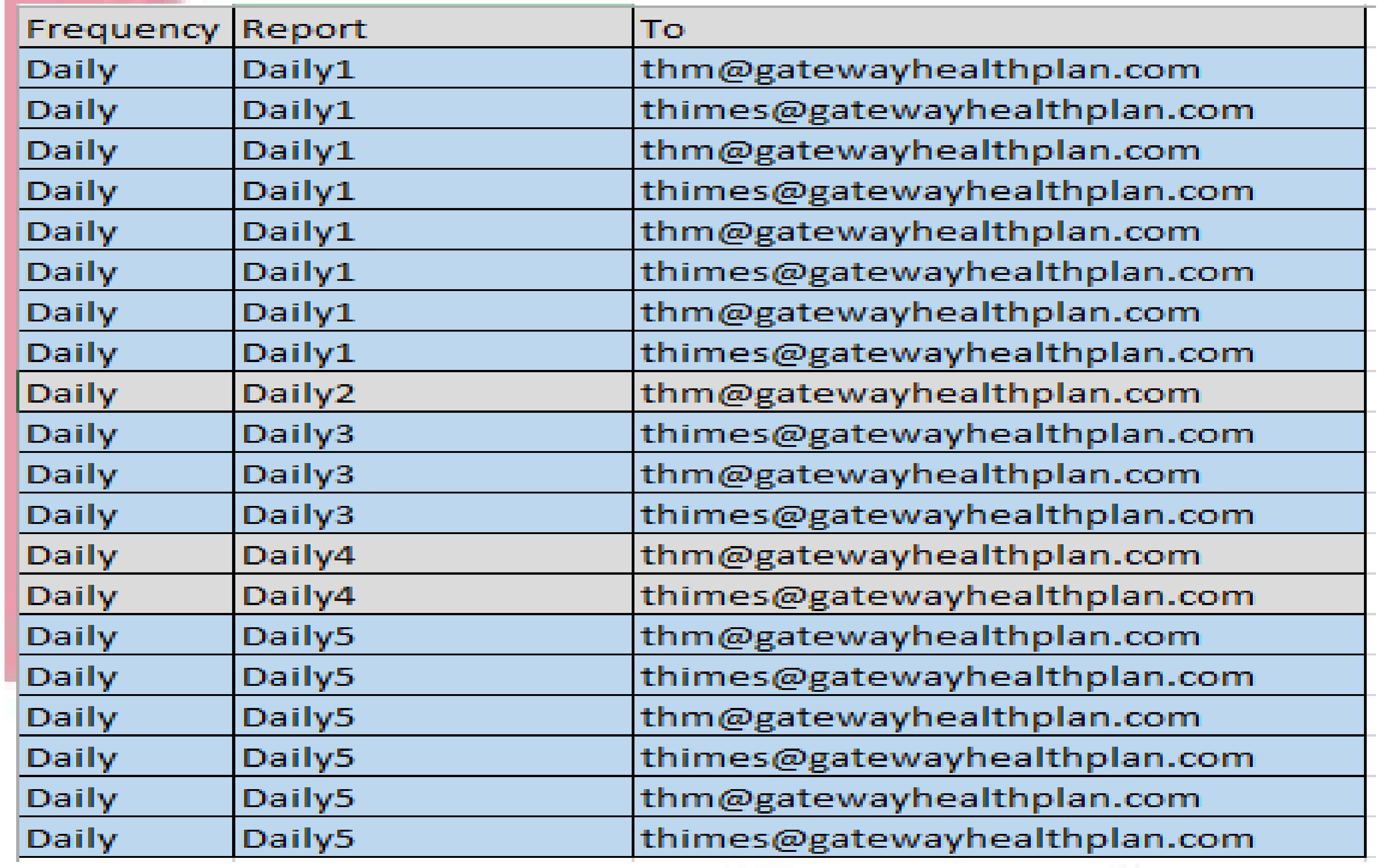


```
data _null_;
    file sendit email
        from=("SASPROD@GATEWAYHEALTHPLAN.COM")
        to=(
            'THM@GatewayHealthPlan.com' 'thimes@GatewayHealthPlan.com' 'THM@GatewayHealthPlan.com'
            'thimes@GatewayHealthPlan.com' 'THM@GatewayHealthPlan.com' 'thimes@GatewayHealthPlan.com'
            )
        subject="Sales - Daily - New Customer Report"
         attach="\\leto\SAS Analyst\thm\New_Customer_Report.XLSX";
        put "Greetings - The Daily New Customer Report is atatched.";
        put;
        put 'Thanks';
        put 'SAS Reporting Team';
run;
```

3. Problems With Standard Email

- You don’t want to alter production code.
- Tedious to add and remove recipients of report
- No centralized area for reports

4. Creating an Email Distribution List

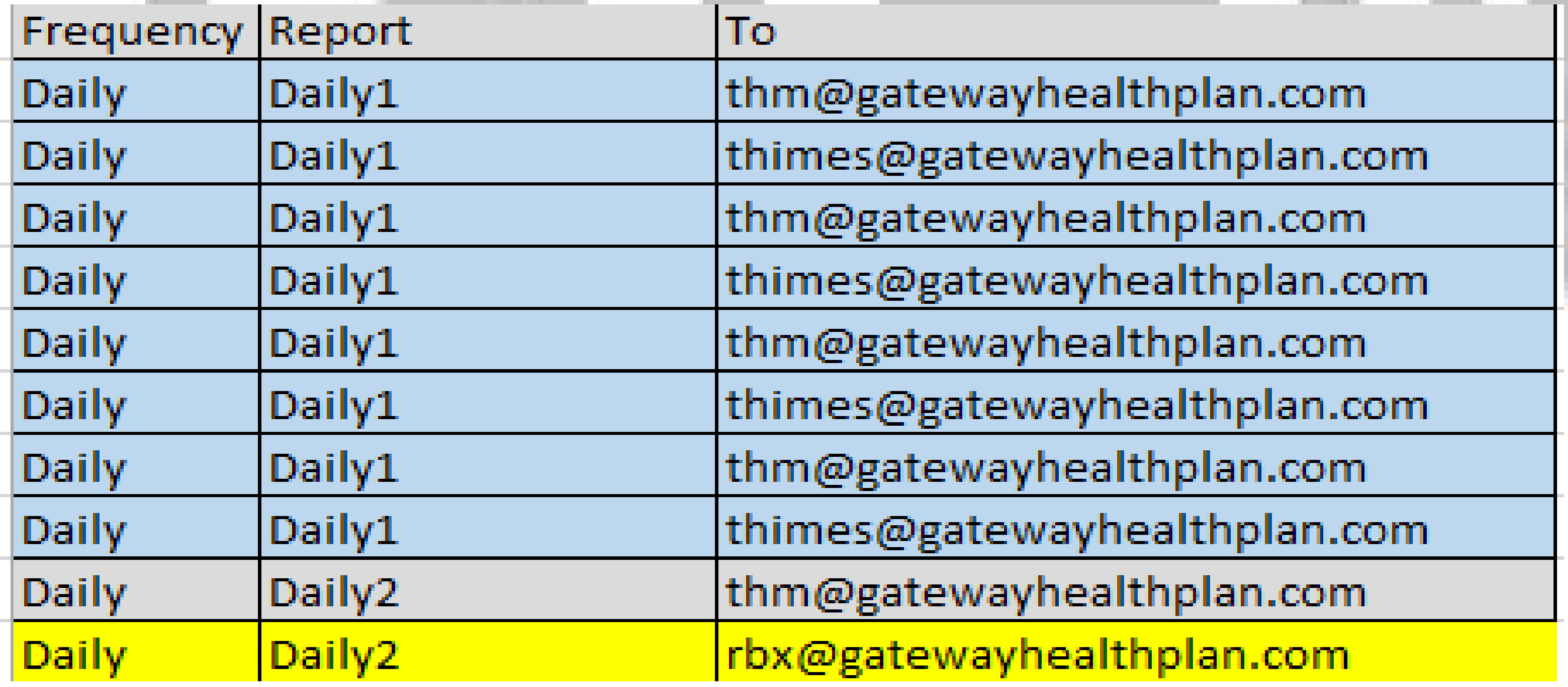


Frequency	Report	To
Daily	Daily1	thm@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily1	thm@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily1	thm@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily1	thm@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily2	thm@gatewayhealthplan.com
Daily	Daily3	thimes@gatewayhealthplan.com
Daily	Daily3	thm@gatewayhealthplan.com
Daily	Daily3	thimes@gatewayhealthplan.com
Daily	Daily4	thm@gatewayhealthplan.com
Daily	Daily4	thimes@gatewayhealthplan.com
Daily	Daily5	thm@gatewayhealthplan.com
Daily	Daily5	thimes@gatewayhealthplan.com
Daily	Daily5	thm@gatewayhealthplan.com
Daily	Daily5	thimes@gatewayhealthplan.com
Daily	Daily5	thm@gatewayhealthplan.com
Daily	Daily5	thimes@gatewayhealthplan.com

5. Advantages Email Distribution List

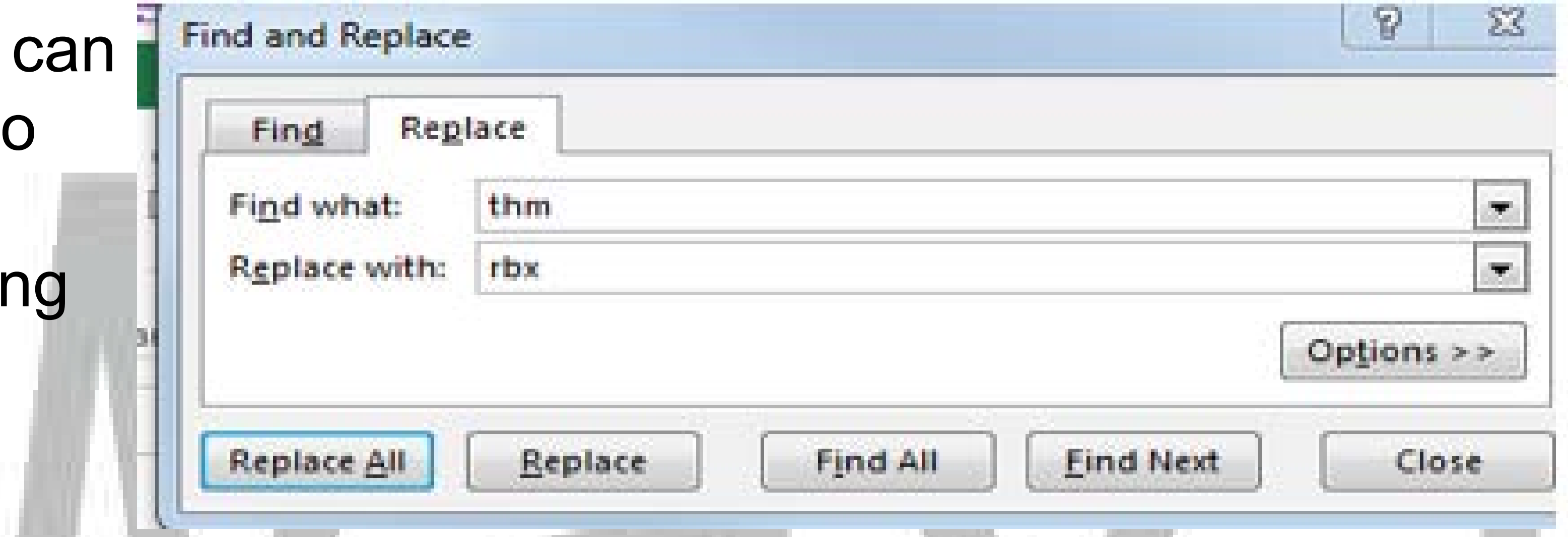
- Any employee who has access to the Excel distribution list can modify the list meaning you don’t require SAS experience to make any changes
- Replacing Recipients in a matter of seconds without touching production code.

Insert Row To Add New Recipient



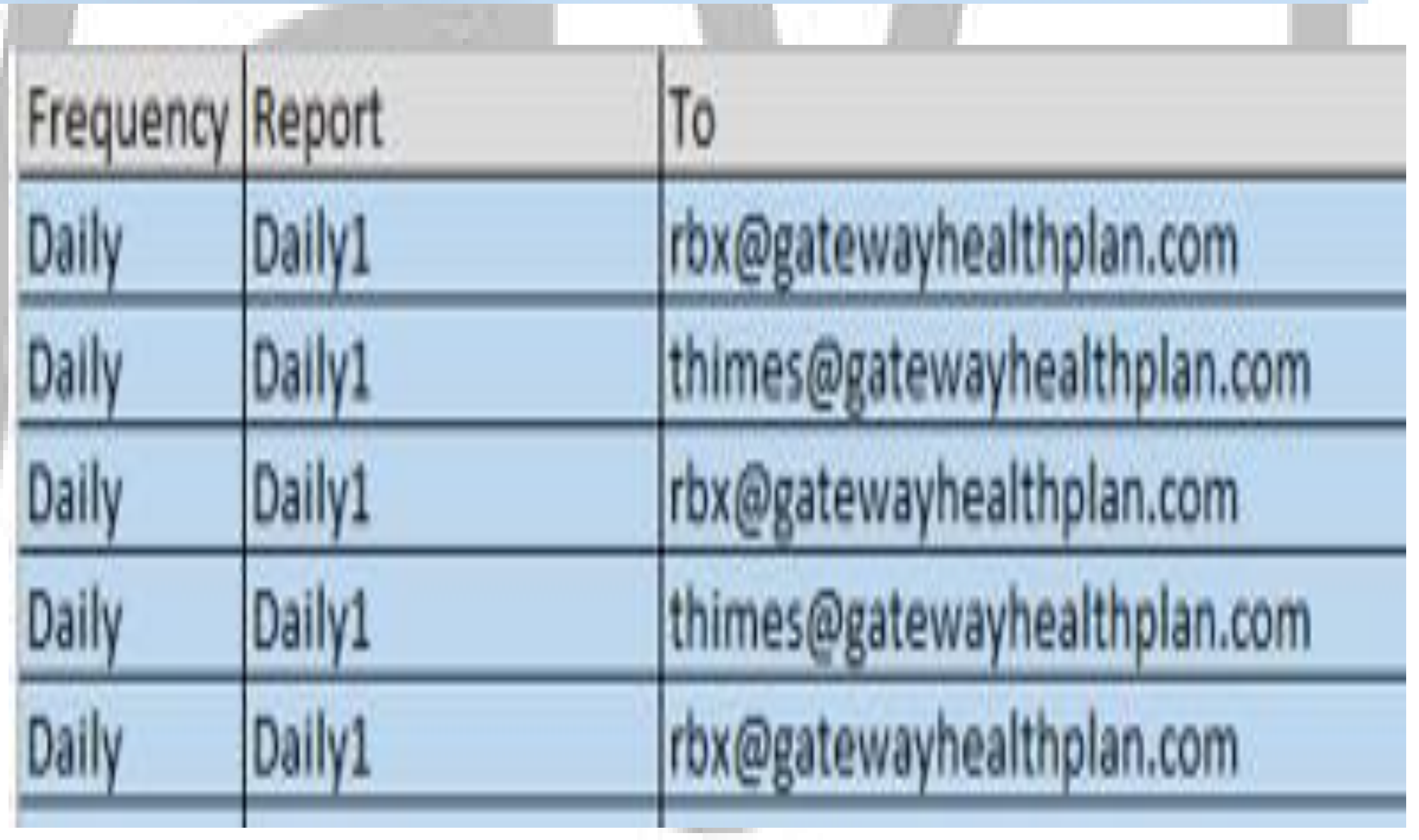
Frequency	Report	To
Daily	Daily1	thm@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily1	thm@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily1	thm@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily1	thm@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily2	thm@gatewayhealthplan.com
Daily	Daily2	rbx@gatewayhealthplan.com

Find and Replace Recipients



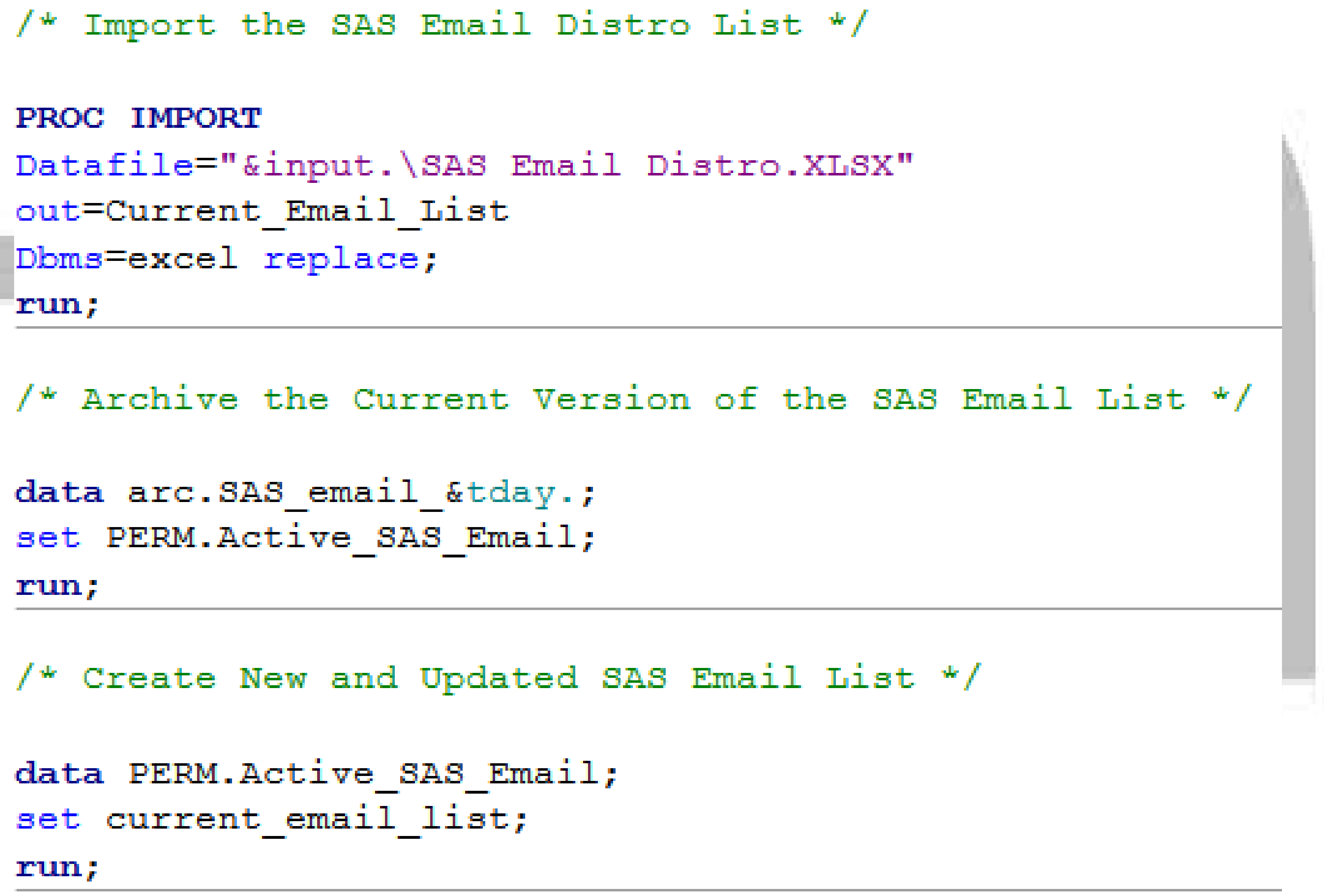
Find what: thm

Replace with: rbx



Frequency	Report	To
Daily	Daily1	rbx@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily1	rbx@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily1	rbx@gatewayhealthplan.com

6. SAS Code to Update Distro List



```
/* Import the SAS Email Distro List */

PROC IMPORT
Datafile="&input.\SAS Email Distro.XLSX"
out=Current_Email_List
Dbms=excel replace;
run;

/* Archive the Current Version of the SAS Email List */

data arc.SAS_email_&tday.;
set PERM.Active_SAS_Email;
run;

/* Create New and Updated SAS Email List */

data PERM.Active_SAS_Email;
set current_email_list;
run;
```

7. Using Email Distro

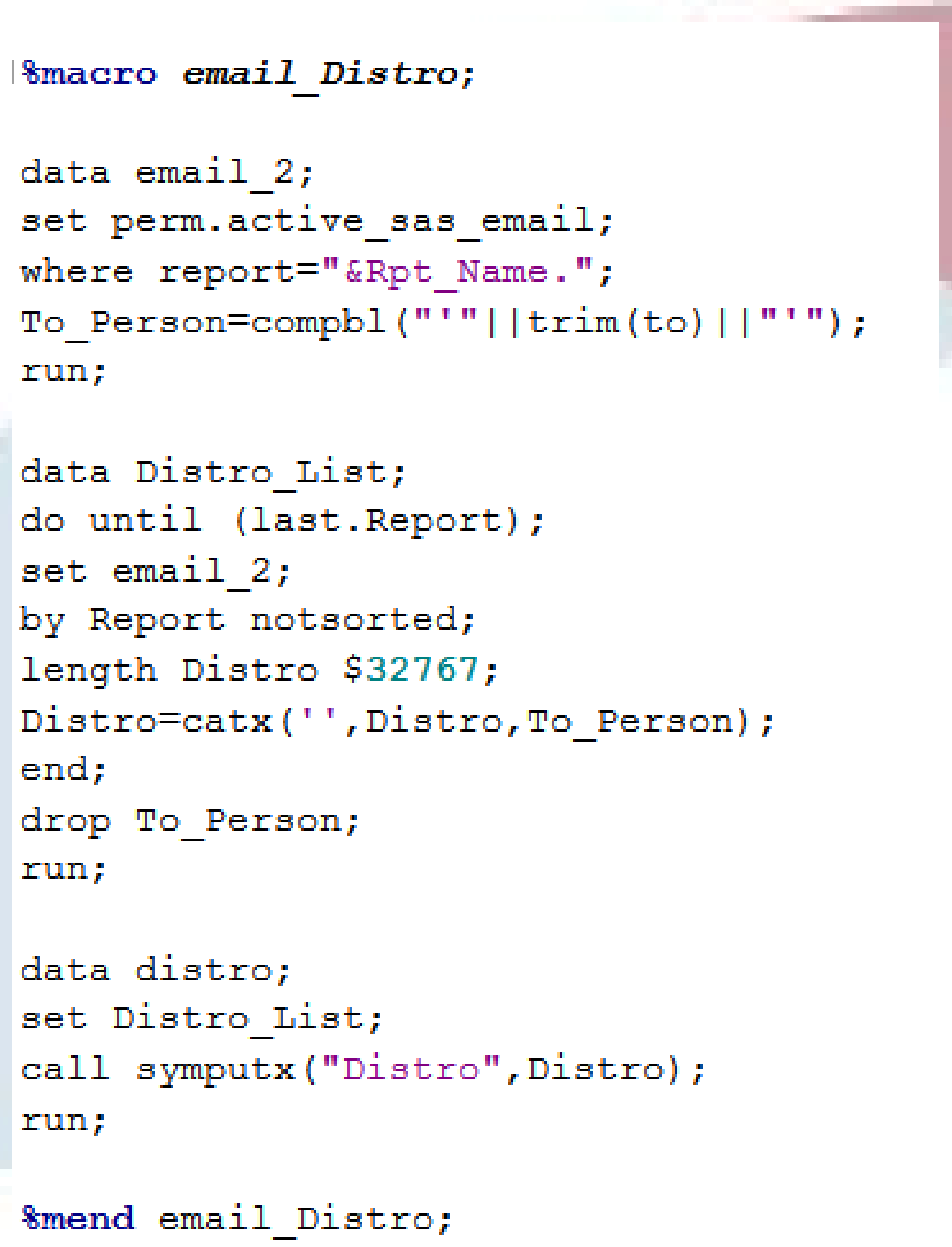
Key Steps:

1. %let RPT_NAME = Daily1
2. %include “C:\EMAIL_DISTRO_LIST.SAS
3. %EMAIL_DISTRO

What they do:

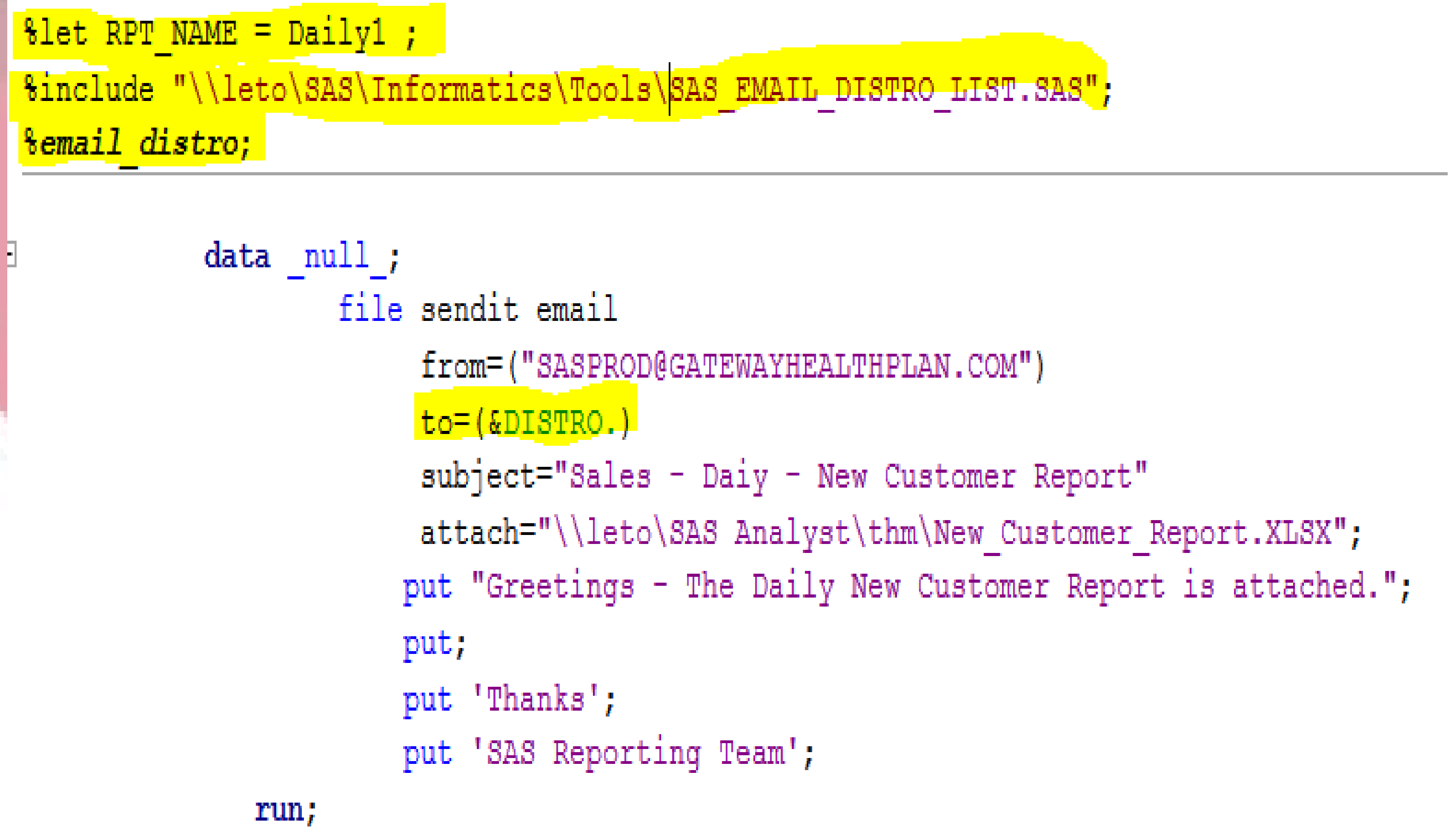
1. Name each report – this should tie to the column “Report” in the Excel or SAS Distribution lists. This is used to identify which users will get the email!
2. Identify the location of the SAS macro EMAIL_DISTRO
3. Call the SAS macro EMAIL_DISTRO

8. Email Disto Macro



```
%macro email_Distro;

data email_2;
set perm.active_sas_email;
where report="&Rpt_Name.";
To_Person=compbl("'"||trim(to)||"'");
run;

data Distro_List;
do until (last.Report);
set email_2;
by Report notsorted;
length Distro $32767;
Distro=catx('',Distro,To_Person);
end;
drop To_Person;
run;

data distro;
set Distro_List;
call symputx("Distro",Distro);
run;

%mend email_Distro;
```

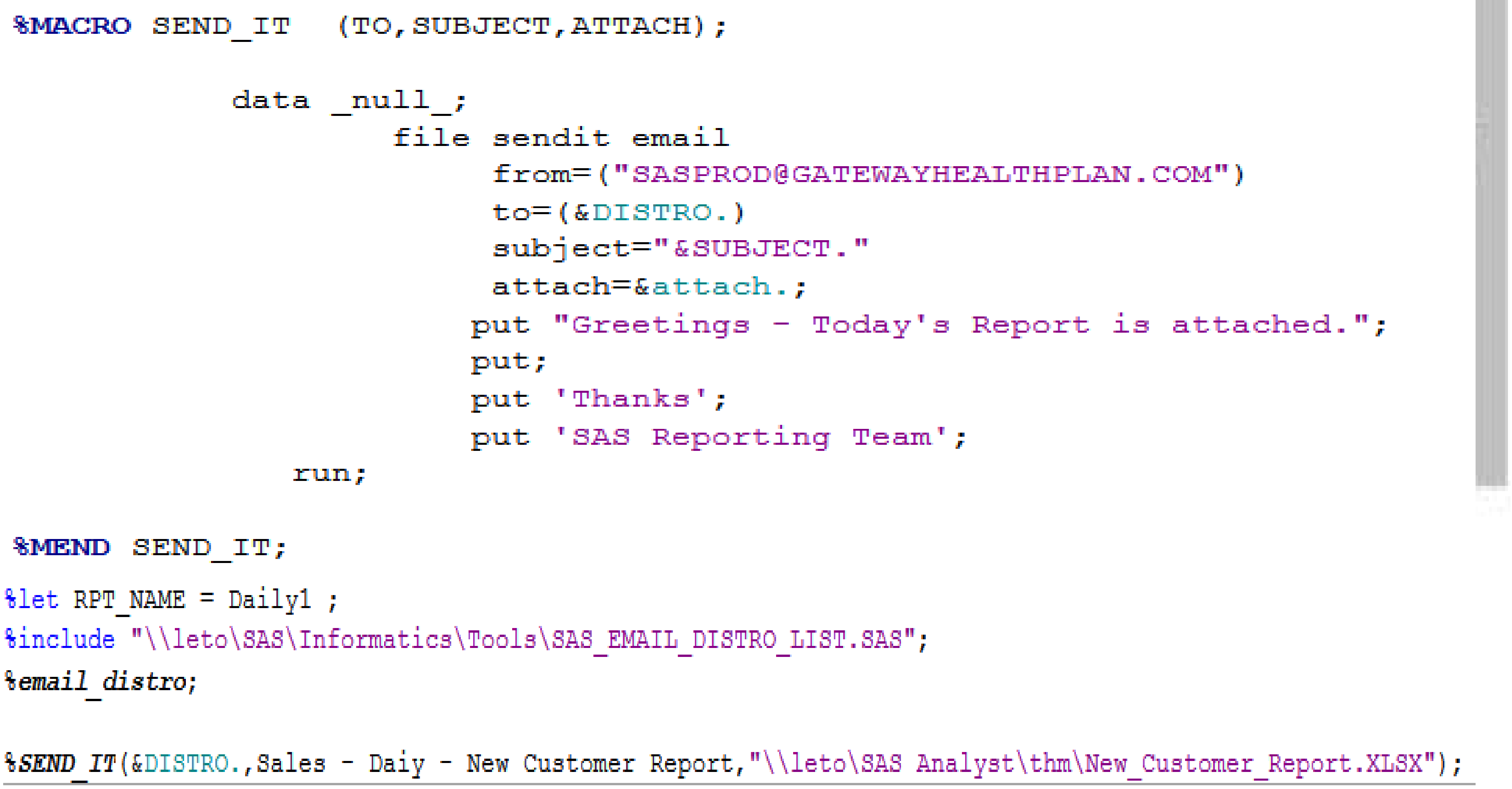
9. New Look At Email Macro



```
%let RPT_NAME = Daily1 ;
%include "\\leto\SAS\Informatics\Tools\SAS_EMAIL_DISTRO_LIST.SAS";
%email_distro;

        data _null_;
             file sendit email
                from=("SASPROD@GATEWAYHEALTHPLAN.COM")
                to=(&DISTRO.)
                subject="Sales - Daiy - New Customer Report"
                attach="\\leto\SAS Analyst\thm\New_Customer_Report.XLSX";
               put "Greetings - The Daily New Customer Report is attached.";
               put;
               put 'Thanks';
               put 'SAS Reporting Team';
          run;
```

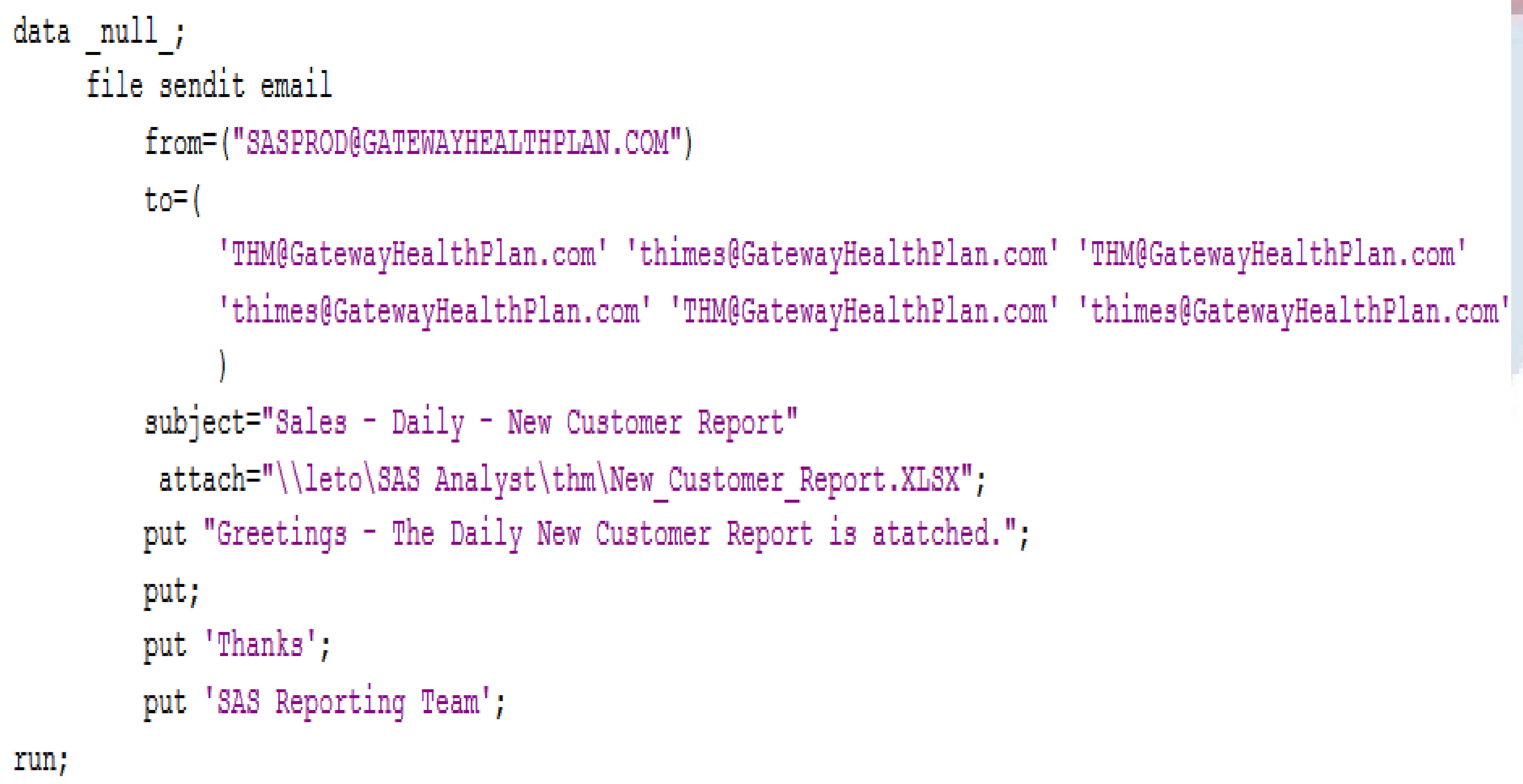
11. Final Recommendation for Enhanced Reporting

The final recommendation to enhance SAS distributing reporting is modifying the above SAS email code into a macro called %SEND_IT which is to be saved outside of the SAS production code.

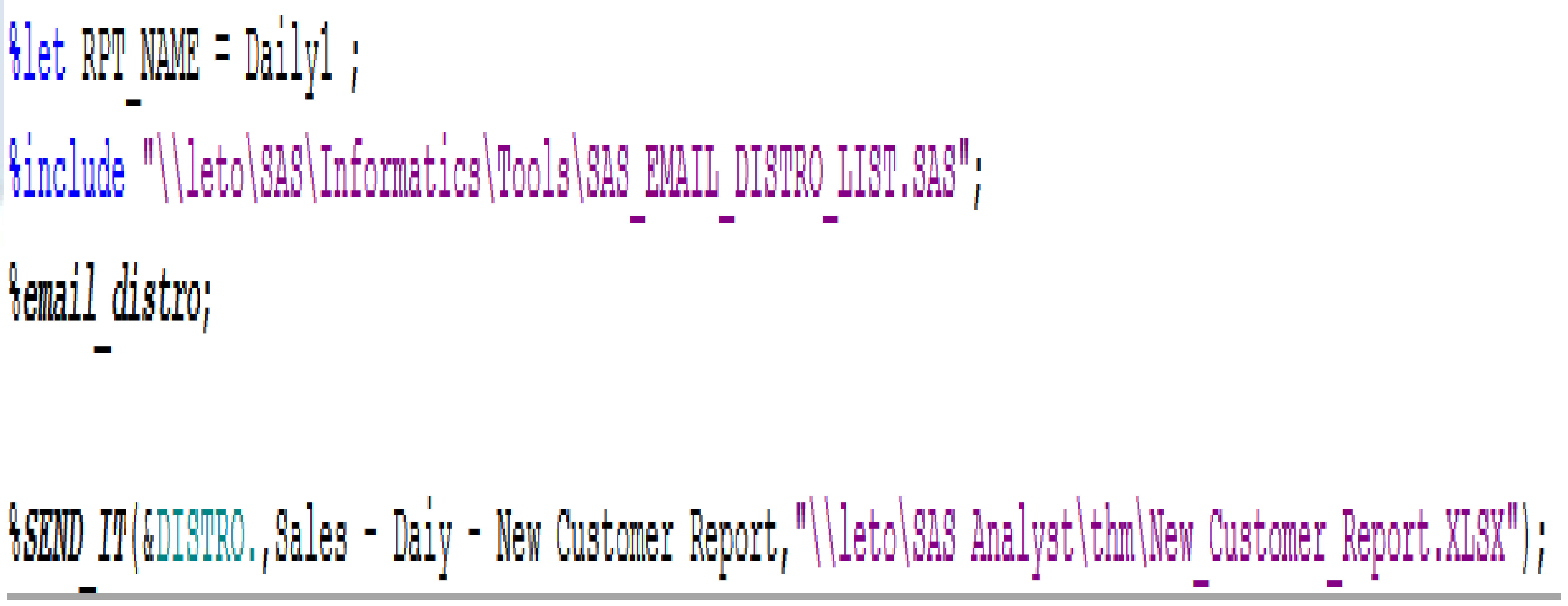


```
%MACRO SEND_IT  (TO,SUBJECT,ATTACH);

          data _null_;
                file sendit email
                     from=("SASPROD@GATEWAYHEALTHPLAN.COM")
                     to=(&DISTRO.)
                     subject="&SUBJECT."
                     attach=&attach.;
                    put "Greetings - Today's Report is attached.";
                    put;
                    put 'Thanks';
                    put 'SAS Reporting Team';
             run;

%MEND SEND_IT;
%let RPT_NAME = Daily1 ;
%include "\\leto\SAS\Informatics\Tools\SAS_EMAIL_DISTRO_LIST.SAS";
%email_distro;

%SEND_IT(&DISTRO.,Sales - Daiy - New Customer Report,"\\leto\SAS Analyst\thm\New_Customer_Report.XLSX");
```

12. Overview (Start – End)



```
data _null_;
     file sendit email
         from=("SASPROD@GATEWAYHEALTHPLAN.COM")
         to=(
              'THM@GatewayHealthPlan.com' 'thimes@GatewayHealthPlan.com' 'THM@GatewayHealthPlan.com'
              'thimes@GatewayHealthPlan.com' 'THM@GatewayHealthPlan.com' 'thimes@GatewayHealthPlan.com'
              )
         subject="Sales - Daily - New Customer Report"
          attach="\\leto\SAS Analyst\thm\New_Customer_Report.XLSX";
         put "Greetings - The Daily New Customer Report is atatched.";
         put;
         put 'Thanks';
         put 'SAS Reporting Team';
run;
```



```
%let RPT_NAME = Daily1 ;
%include "\\leto\SAS\Informatics\Tools\SAS_EMAIL_DISTRO_LIST.SAS";
%email_distro;


%SEND_IT(&DISTRO.,Sales - Daiy - New Customer Report,"\\leto\SAS Analyst\thm\New_Customer_Report.XLSX");
```

13. Conclusion

When creating a SAS production environment it is best practice to touch the SAS code that lives there as little as possible. While using the standard SAS Email Code will deliver your reports to the correct individuals it is often tedious to modify this code especially if it involves removing or adding users in large volume. Also, this approach does not involve keeping a SAS Distribution List which can be very beneficial when keeping a reporting inventory. Lastly, making changes to this SAS Distribution list is much easier than modifying the standard SAS Email Code. By creating a SAS Distribution list and SAS email macro which both live outside of your SAS production environment you are ensuring stability in the success rate of your production code and also saving yourself time when modifying distribution lists while keeping best practices.

Clean Distribution and Emails:
A Reporting Dream

Ryan Bixler and Travis Himes, Gateway Health Plan®, Pittsburgh, PA

ABSTRACT

In a perfect world of reporting, all code is automated and successfully invoked without manual intervention or ever touching the SAS® production code. Unfortunately, this is not the case for all reports across a company. This paper demonstrates how to efficiently add/remove recipients to a SAS report distribution list without manipulating the SAS production code. To do so my coworker and I developed an email distribution list along with a macro to send SAS reports. Together, these two macros provide a solution to efficient reporting and also create a report inventory. Using these macros allows users to get away from altering SAS production code to simply add/remove users who receive a report which is a best practice for most companies. We plan to educate users on how to develop and use these macros efficiently.

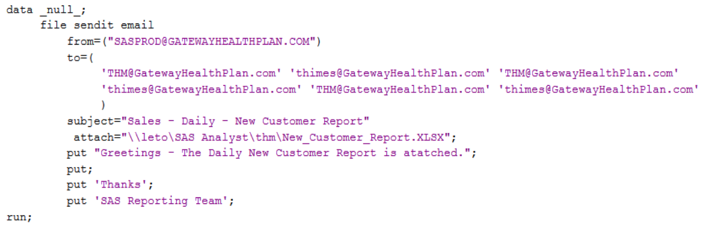
INTRODUCTION

Suppose you work at a company with thousands of employees who utilize hundreds of regular scheduled SAS reports (daily, weekly, monthly, etc.) to make their key business decisions and you are responsible for ensuring the correct reports go to the correct people. Now also picture this same company being one which is continuously growing and expanding its workforce while at the same time seeing leadership and turnover changes throughout the organization. Manually editing SAS production code each time a new employee needs added to a report distribution list while another employee needs removed from a completely separate report could leave you spending countless hours each week manipulating and editing production code which, in best practice, should remain untouched. Additionally, manually inserting and removing email addresses can be stressful and one simple typo can cause an entire report to fail on something which can easily be avoided with the use of a few simple macros. To avoid such tedious tasks taking large amounts of time and jeopardizing the success rate of a production SAS code it is crucial to maintain an accurate reporting distribution list and also have an email macro which both live outside of SAS production code.

This paper demonstrates how to send emails from SAS 9.4 via SAS Enterprise Guide 7.1 to my Microsoft Outlook 2013 email address with standard SAS email code and also shows the advantages of using a SAS email macro which relies on a reporting distribution list.

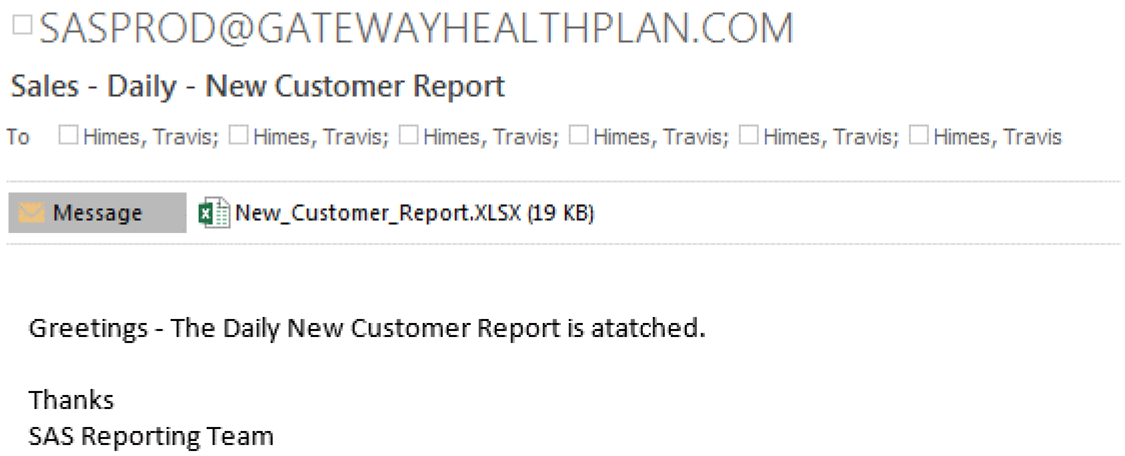
STANDARD SAS EMAIL CODE

Below is some standard SAS code which demonstrates how to send an Excel report called New_Customer_Report from SAS to a group of individuals. In this example, all the email addresses are my personal work email as I didn't want to overload and confuse my coworkers. For the purpose of this paper please pretend each email is a unique person and keep this mindset throughout.



```
data _null_;
     file sendit email
         from=("SASPROD@GATEWAYHEALTHPLAN.COM")
         to=(
              'THM@GatewayHealthPlan.com' 'thimes@GatewayHealthPlan.com' 'THM@GatewayHealthPlan.com'
              'thimes@GatewayHealthPlan.com' 'THM@GatewayHealthPlan.com' 'thimes@GatewayHealthPlan.com'
              )
         subject="Sales - Daily - New Customer Report"
          attach="\\leto\SAS Analyst\thm\New_Customer_Report.XLSX";
         put "Greetings - The Daily New Customer Report is atatched.";
         put;
         put 'Thanks';
         put 'SAS Reporting Team';
run;
```

This code runs successfully with no errors and is delivered to all the email addresses listed above. Below is an example of the email received in outlook.



□ SASPROD@GATEWAYHEALTHPLAN.COM

Sales - Daily - New Customer Report

To □ Himes, Travis; □ Himes, Travis; □ Himes, Travis; □ Himes, Travis; □ Himes, Travis; □ Himes, Travis

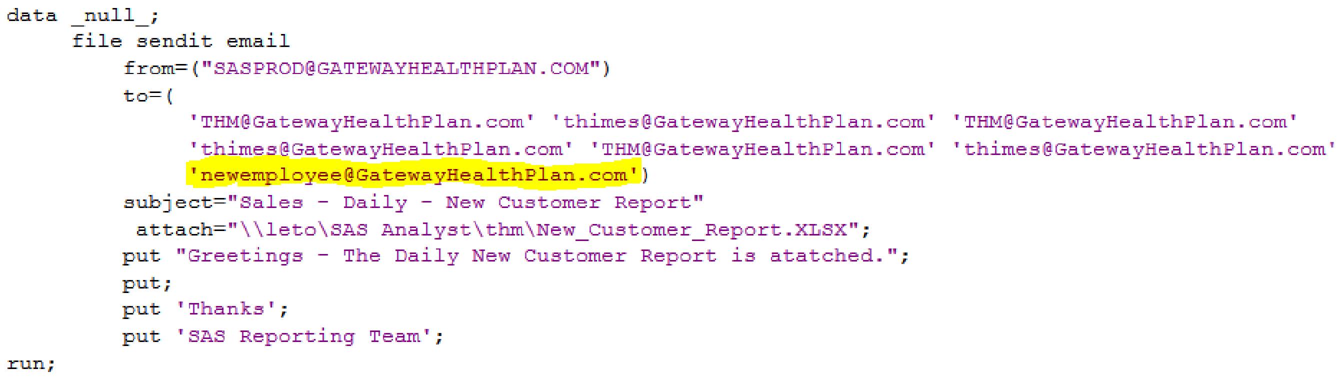
New_Customer_Report.XLSX (19 KB)

Greetings - The Daily New Customer Report is atatched.

Thanks
SAS Reporting Team

PROBLEMS WITH USING STANDARD SAS EMAIL CODE

The standard SAS email code above will deliver the reports to the right people if coded correctly. However, all organizations will regularly go through change and need employees to be added and/or removed from certain reports. For example, if a new employee is hired and needs added to the above report we could simply make the following addition to the above code. In this example the employee to be added is named New Employee.



```
data _null_;
     file sendit email
         from=("SASPROD@GATEWAYHEALTHPLAN.COM")
         to=(
              'THM@GatewayHealthPlan.com' 'thimes@GatewayHealthPlan.com' 'THM@GatewayHealthPlan.com'
              'thimes@GatewayHealthPlan.com' 'THM@GatewayHealthPlan.com' 'thimes@GatewayHealthPlan.com'
              'newemployee@GatewayHealthPlan.com')
         subject="Sales - Daily - New Customer Report"
          attach="\\leto\SAS Analyst\thm\New_Customer_Report.XLSX";
         put "Greetings - The Daily New Customer Report is atatched.";
         put;
         put 'Thanks';
         put 'SAS Reporting Team';
run;
```

This will cause the report to go to the new employee and also the emails which already were on the distribution list, however, there a few issues with this process. The main issue is this approach involves altering SAS production code and risking the chance that the program would fail due to a programmer incorrectly making the above change. For example, if the programmer forgets to simply add a single quote around the new email address or places a semi colon in the wrong section the report will not be sent to the distribution list. Most importantly, it is best practice in any industry to modify SAS production code as little as possible. Of course there are certain exceptions to this rule like a business requirement which could alter the logic of a report or a major change to a database, but modifying the reporting email list does not fall into this category. If we use the process above there is a high likelihood we would be modifying production code numerous times as each time a new employee needs added or removed from the report we would have to modify the email section of the code.

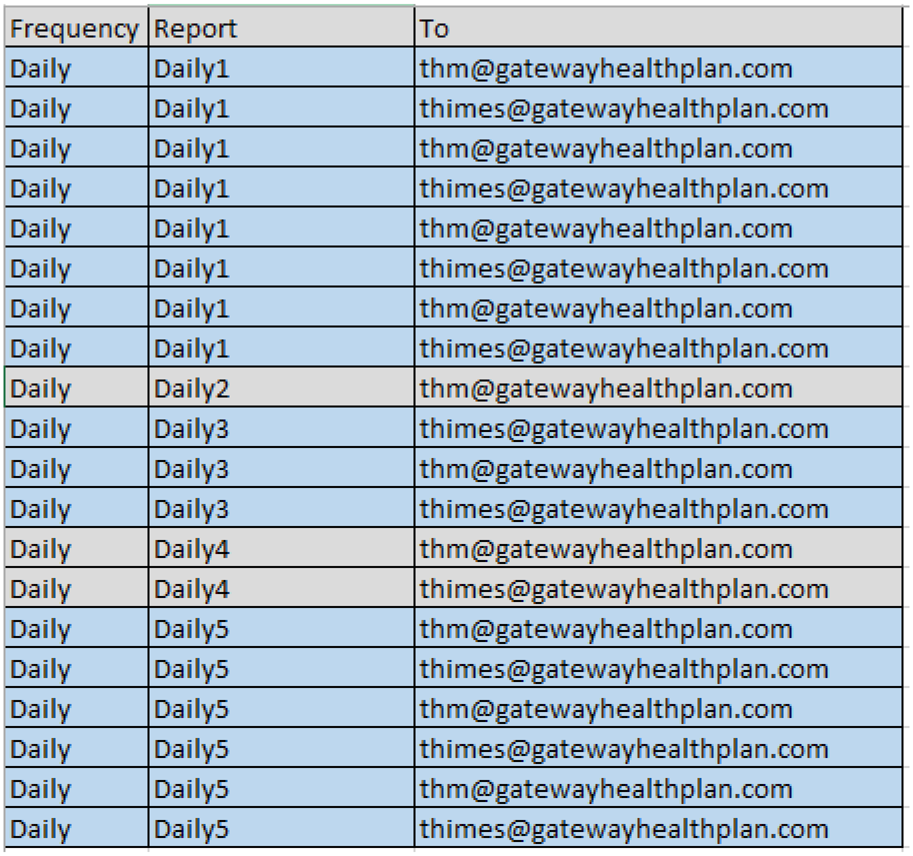
The biggest issue with this email approach would come from a situation where a programmer was asked to add a new employee to all the reports another employee is on. Similarly, to replace employee A with employee B on all reports or remove an employee from all SAS reports. This would be very time consuming as we would need to open up every single SAS production code and make the requested changes which as mentioned above is extremely risky and not best practice. Imagine a company with hundreds of production codes and having to open each SAS report to modify the users receiving the reports in the email section. This issue arises because using the code above does not identify a distribution list meaning we cannot easily see which users are receiving which reports unless we open the SAS source code.

CREATING AN EMAIL DISTRIBUTION LIST

The proposed solution to avoid the potential coding headaches described above is to scratch the standard SAS code approach and instead develop a SAS email distribution list and a SAS email macro.

The first step is developing a SAS email distribution list. This paper proposes two ways to develop these distribution lists. Whichever approach is used the idea is to make these distribution lists easily accessible and to ensure they can be easily modified. The first approach is creating the list in Excel. Below is an example of the distribution lists for 5 separate SAS reports. In this example all 5 of the reports are daily and go to my email address. This is done for simplicity but demonstrate the process effectively. The “Frequency” is simply when the report runs (daily, monthly, yearly). This is a useful column as if you are ever asked to list all the reports your team runs daily you could easily find the answer here. The “Report” column is the name of the report and the “To” column includes the entire distribution list of employees receiving the report. These will be described in more detail later.

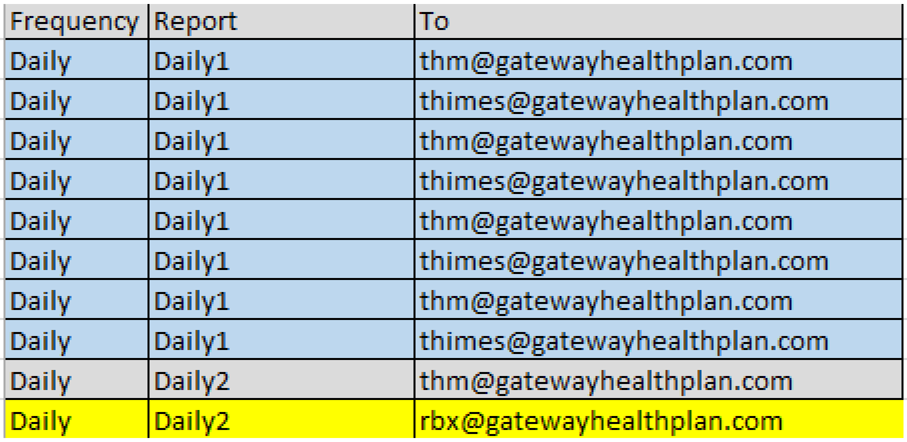
Here is a snapshot of the Excel Distribution List.



Frequency	Report	To
Daily	Daily1	thm@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily1	thm@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily1	thm@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily1	thm@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily2	thm@gatewayhealthplan.com
Daily	Daily3	thimes@gatewayhealthplan.com
Daily	Daily3	thm@gatewayhealthplan.com
Daily	Daily3	thimes@gatewayhealthplan.com
Daily	Daily4	thm@gatewayhealthplan.com
Daily	Daily4	thimes@gatewayhealthplan.com
Daily	Daily5	thm@gatewayhealthplan.com
Daily	Daily5	thimes@gatewayhealthplan.com
Daily	Daily5	thm@gatewayhealthplan.com
Daily	Daily5	thimes@gatewayhealthplan.com
Daily	Daily5	thm@gatewayhealthplan.com
Daily	Daily5	thimes@gatewayhealthplan.com

There are two big advantages to keeping the distribution list in Excel. The first advantage is any employee who has access to the Excel distribution list can modify the list meaning you don’t require SAS experience to make any changes. Granting access to only the correct employees is crucial to ensure the accuracy of the distribution list.

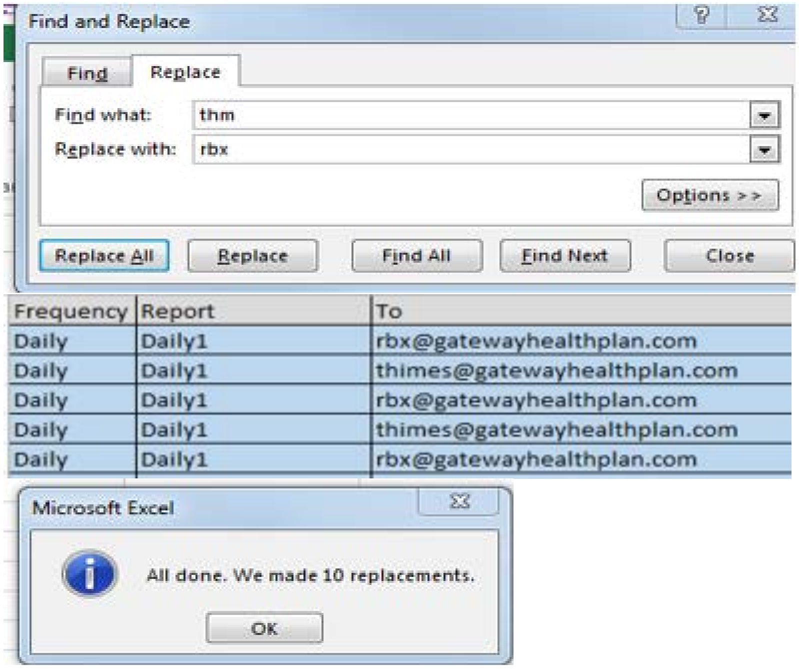
The biggest advantage comes when making changes to the distribution of a specific report. For example, say you have been tasked with adding user RBX to the report “Daily2”. This is a simple request which can be done by inserting a new row in the Excel and putting the information shown below.



Frequency	Report	To
Daily	Daily1	thm@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily1	thm@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily1	thm@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily1	thm@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily2	thm@gatewayhealthplan.com
Daily	Daily2	rbx@gatewayhealthplan.com

In the standard SAS code example we would have to open the SAS production code for Daily2 and modify the SAS production code to add user RBX's email to the report. Remember, this is not best practice and could jeopardize the next run of Daily2.

Additionally, say the employee THM has been fired and needs to be replaced by his more intelligent and better looking successor RBX. In the standard SAS example we would have to open every single SAS production code, check to see if THM was on the email list, and replace THM with RBX when appropriate. This could take hours for organizations with hundreds of SAS production codes. Imagine the amount of time wasted if THM was only on 1 report and the organization had 500+ reports. However, by using the Excel SAS Email distribution list this can be done in a matter of seconds! Simply hit "CTRL+F" and "Replace" "THM" with "RBX" and click "replace all" and BOOM! The screenshots that follow demonstrate this process. The full Excel Distribution list was not included for size purposes.



Find what: thm

Replace with: rbx

Frequency	Report	To
Daily	Daily1	rbx@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily1	rbx@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily1	rbx@gatewayhealthplan.com

All done. We made 10 replacements.

In this small example we replaced THM with RBX on 10 reports in a matter of seconds without touching any SAS production code and avoided countless time opening reports in which THM wasn't even on the distribution list. Again, this is just a small example but imagine if THM was on 300 separate reports!

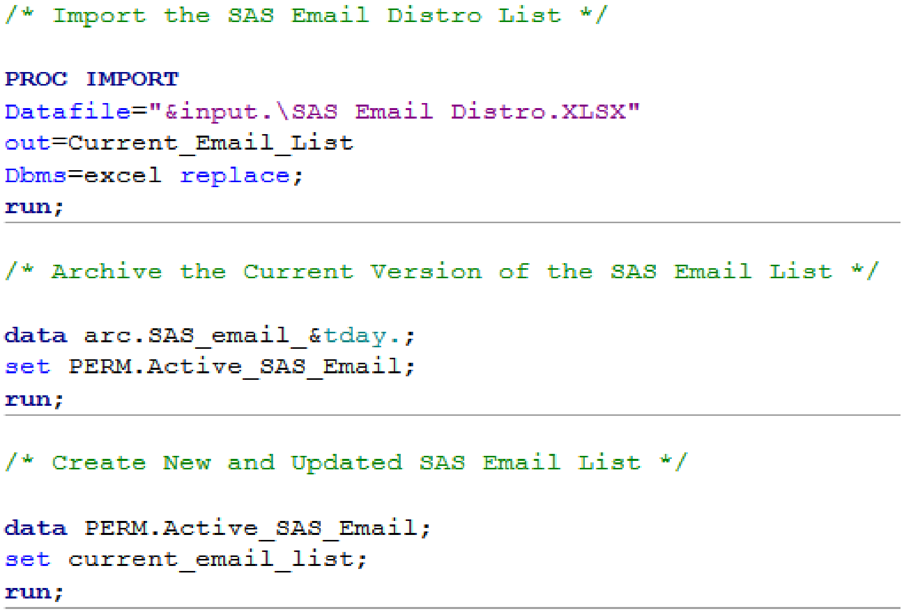
The Excel Distribution list is great and can be easily modified but this isn't the end of our solution as we haven't shown how to tie this Excel list to the SAS email logic.

SAS DATASET EMAIL DISTRIBUTION LIST

One red flag you may have thought of while reading about the Excel Email Distribution list is what if something happens to the distribution list. For example, what if someone deletes the Excel file or accidentally puts in wrong information and corrupts the file? All the hard work and time spent into our Excel Distribution list will be gone unless someone in IT could restore the file. What if we could avoid this situation?

The proposed second solution to improving the standard SAS email code is to also create the distribution list in SAS. This is done via a SAS program which imports the above Excel file, saves the current Excel as a permanent SAS dataset, and also archives the prior version of the Excel so any errors or corruption the Excel file can be easily fixed. Our company has this code run daily so we always have an updated reporting distribution list for the following day and also archive the prior day's list as a backup plan. It is up to the user to determine how frequent to run this code. If the distribution lists don't change frequently you can move this to weekly or even an ad-hoc basis (only when a change is made).

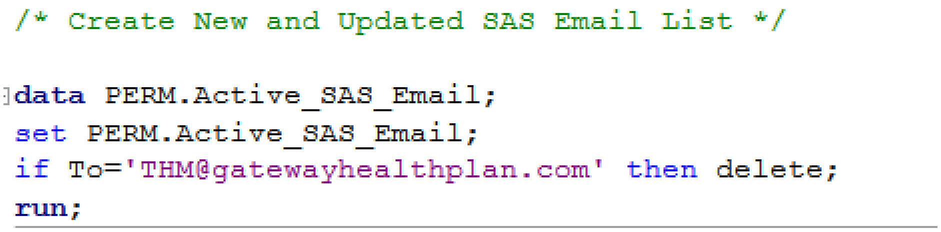
Here is the SAS code which creates the SAS distribution list in a permanent SAS table saved at the location we identify as PERM while also archiving yesterday's file in the location ARC.



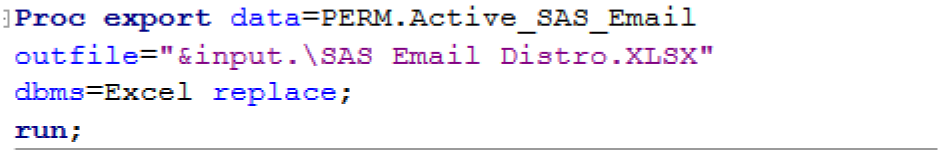
```
/* Import the SAS Email Distro List */

PROC IMPORT
Datafile="&input.\SAS Email Distro.XLSX"
out=Current_Email_List
Dbms=excel replace;
run;

/* Archive the Current Version of the SAS Email List */

data arc.SAS_email_&tday.;
set PERM.Active_SAS_Email;
run;

/* Create New and Updated SAS Email List */

data PERM.Active_SAS_Email;
set current_email_list;
run;
```

Now we can avoid any disasters if our Excel file goes wrong by opening the most successful archived SAS email distribution dataset.

Also, as a SAS professional it would almost be treasonous not to mention that any changes to the distribution list could be made through SAS and without the use of the Excel example above. Say user THM has been fired for napping on the job and needs to be removed from all SAS production jobs. Some basic SAS code below could do the trick with leaving Excel in the dust. The code below shows we can simply delete user THM from any report with a simple delete statement and then export the SAS Distribution list to overwrite the Excel Distribution list to ensure the two lists are identical.

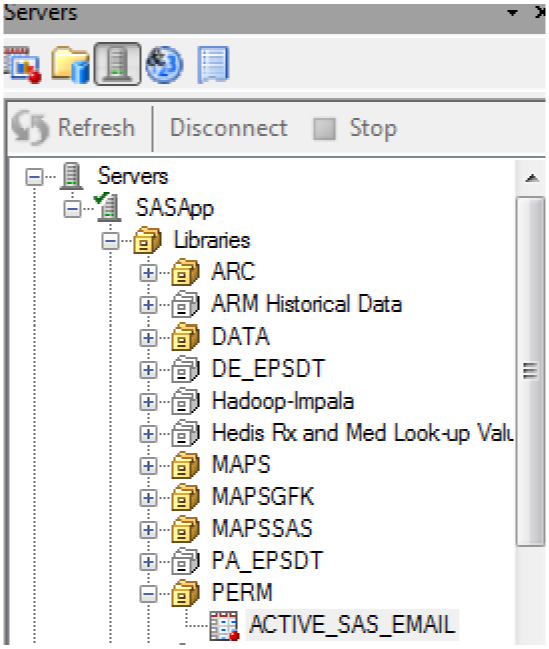


```
/* Create New and Updated SAS Email List */

data PERM.Active_SAS_Email;
set PERM.Active_SAS_Email;
if To='THM@gatewayhealthplan.com' then delete;
run;
```



```
Proc export data=PERM.Active_SAS_Email
outfile="&input.\SAS Email Distro.XLSX"
dbms=Excel replace;
run;
```

Also, since we are using SAS Enterprise Guide there is a simple “point and click” way to add or remove users from the SAS distribution list. By making the following selections: Servers => SASApp => Libraries => Perm => ACTIVE_SAS_EMAIL you can open up the SAS Distribution list.



Once open you can click EDIT => (uncheck) PROTECT DATA and make any changes to the distribution list. Just be sure to click EDIT => (check) PROTECT DATA and save as a permanent dataset again when you are finished and export the updated dataset to Excel to ensure consistency.

Whether you update the distribution list in Excel or SAS the main takeaway is we are following best practice by updating the distribution lists of SAS production reports without touching SAS production code.

USING YOUR EMAIL DISTRIBUTION LIST

Congratulations! We now have our Email Distribution lists ready to go. Next, we must demonstrate how to use this SAS Email Distribution list to send out reports to the correct users. This can be done by creating another SAS macro we call EMAIL_DISTRO.

Before using EMAIL_DISTRO you must first ensure a few pieces of key SAS code are in all your production code. It is best practice to use a standard SAS template to avoid forgetting these 3 key steps which are the following:

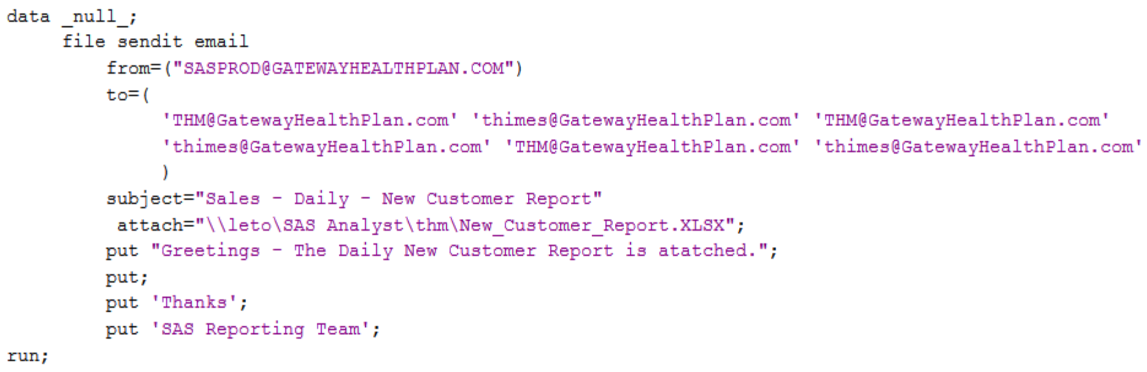
1. %let RPT_NAME = Daily1;
2. %include “C:\EMAIL_DISTRO_LIST.SAS;
3. %EMAIL_DISTRO;

These 3 key pieces of code will:

1. Name each report – this should tie to the column “Report” in the Excel or SAS Distribution lists. <u>This is used to identify which users will get the email!</u>
2. Identify the location of the SAS macro EMAIL_DISTRO
3. Call the SAS macro EMAIL_DISTRO

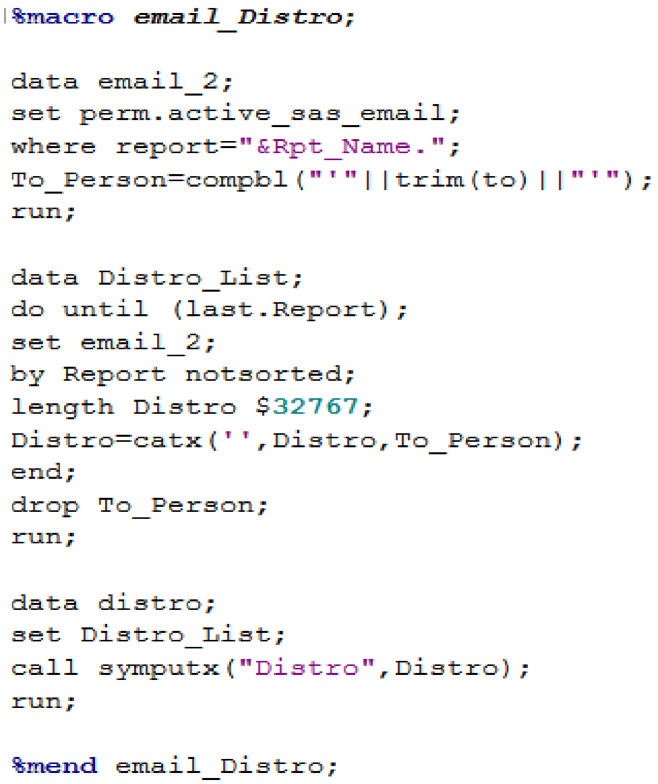
Again, the 3 lines of code above are required in all production code to ensure any report on the Excel and/or SAS Email Distribution list is emailed properly to the correct list.

The EMAIL_DISTRO itself is a simple piece of SAS code and is listed at the top of the next page due to size restrictions. Moving forward, the macro simply uses the current SAS Email Distribution list which we created above called PERM.Active_SAS_EMAIL and only pulls the rows with the associated value entered in the newly created macro variable RPT_NAME (Key Step 1 from above). Also, the first step creates a new variable called “To_Person” which is identical to the email addresses in “To” but adds a single quote around each email address. As we saw above (and again below) in the Standard SAS Email Code all email addresses require a single parenthesis around them.



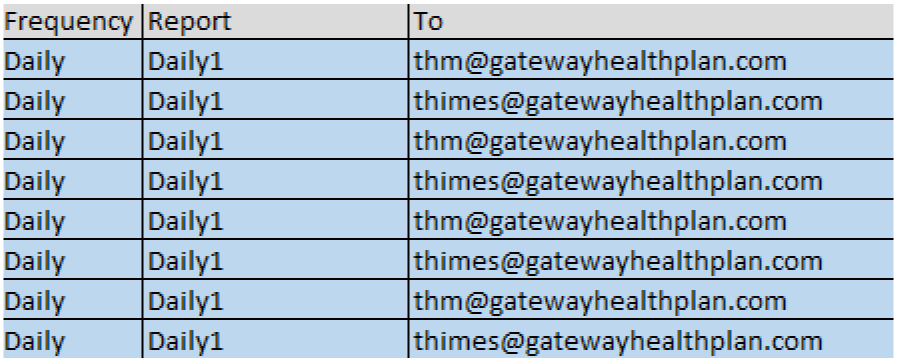
```
data _null_;
     file sendit email
         from=("SASPROD@GATEWAYHEALTHPLAN.COM")
         to=(
              'THM@GatewayHealthPlan.com' 'thimes@GatewayHealthPlan.com' 'THM@GatewayHealthPlan.com'
              'thimes@GatewayHealthPlan.com' 'THM@GatewayHealthPlan.com' 'thimes@GatewayHealthPlan.com'
              )
         subject="Sales - Daily - New Customer Report"
          attach="\\leto\SAS Analyst\thm\New_Customer_Report.XLSX";
         put "Greetings - The Daily New Customer Report is atatched.";
         put;
         put 'Thanks';
         put 'SAS Reporting Team';
run;
```

Below is the macro EMAIL_DISTRO we are discussing.



```
%macro email_Distro;

data email_2;
set perm.active_sas_email;
where report="&Rpt_Name.";
To_Person=compbl("'"||trim(to)||"'");
run;

data Distro_List;
do until (last.Report);
set email_2;
by Report notsorted;
length Distro $32767;
Distro=catx(' ',Distro,To_Person);
end;
drop To_Person;
run;

data distro;
set Distro_List;
call symputx("Distro",Distro);
run;

%mend email_Distro;
```

This means for the Daily Report called “Daily1” this macro will pull the following records which are all the employees who should receive this report:



Frequency	Report	To
Daily	Daily1	thm@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily1	thm@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily1	thm@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com
Daily	Daily1	thm@gatewayhealthplan.com
Daily	Daily1	thimes@gatewayhealthplan.com

Again, the data step EMAIL_2 will create “TO_PERSON” a field which mirrors “TO” but with single quotes around each email address. This means that thm@gatewayhealth.com (TO) will become ‘thm@gatewayhealthplan.com’ (TO_PERSON).

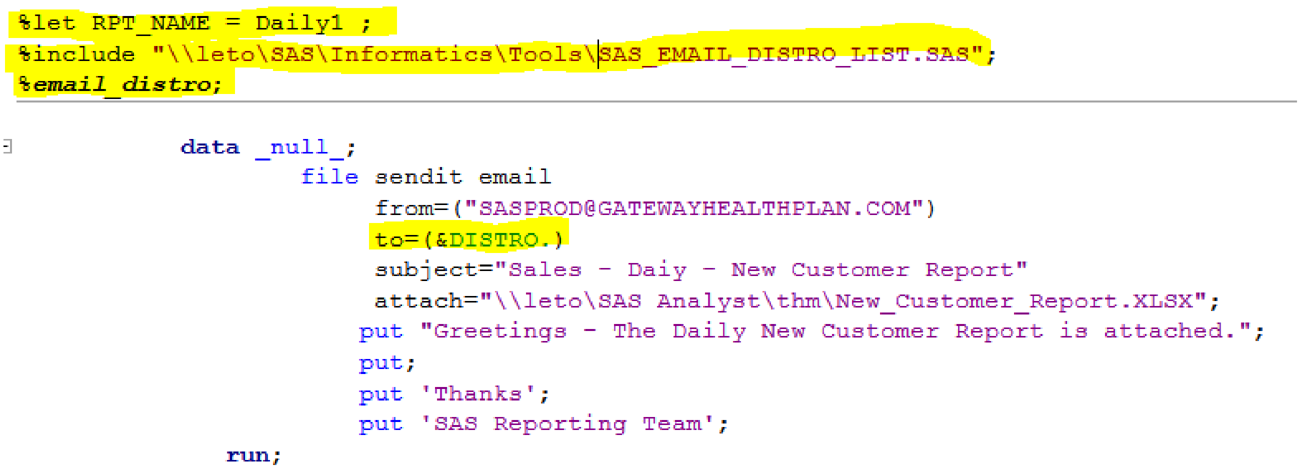
The DISTRO_LIST data step creates a new variable called DISTRO which is one long text field of all the values in “TO_PERSON”. This means this variable DISTRO has all the email addresses intended to receive the specific SAS report (Daily1 in this case) in a single variable. For this report, DISTRO = ‘thm@gatewayhealthplan.com’ ‘thimes@gatewayhealthplan.com’ ‘thm@gatewayhealthplan.com’ , etc.

DISTRO is the key variable created which will be used to ensure all SAS reports are going to the correct distribution list. It becomes a macro variable &DISTRO in the last data step (appropriately named DISTRO) in %EMAIL_DISTRO.

Together the macro variables &RPT_NAME and &DISTRO along with a correctly maintained SAS/Excel Reporting Distribution list control the fate of our SAS Prod environment.

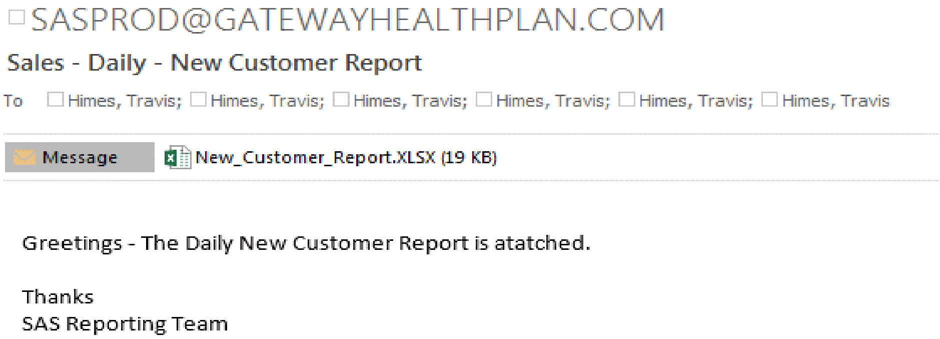
SAS EMAIL MACRO

Now that we have &RPT_NAME and &DISTRO we will now demonstrate how to use them in the standard SAS email code and also modify the standard SAS email code to be a macro.



```
%let RPT_NAME = Daily1 ;
%include "\\leto\SAS\Informatics\Tools\SAS_EMAIL_DISTRO_LIST.SAS";
%email_distro;

            data _null_;
                file sendit email
                     from=("SASPROD@GATEWAYHEALTHPLAN.COM")
                     to=(&DISTRO.)
                     subject="Sales - Daiy - New Customer Report"
                     attach="\\leto\SAS Analyst\thm\New_Customer_Report.XLSX";
                    put "Greetings - The Daily New Customer Report is attached.";
                    put;
                    put 'Thanks';
                    put 'SAS Reporting Team';
               run;
```

By adding the 3 key pieces of code we mentioned last section and using &DISTRO in the TO part of the SAS email we have made emailing SAS reports much easier with less risk of failing. As we have stressed multiple times any changes to the distribution list can be made either in the Excel list or the SAS dataset and the SAS production code will remain untouched by simply having &DISTRO to take place of all the employees receiving the SAS report. The result in Outlook 2013 is the same as the standard SAS code.



□ SASPROD@GATEWAYHEALTHPLAN.COM

Sales - Daily - New Customer Report

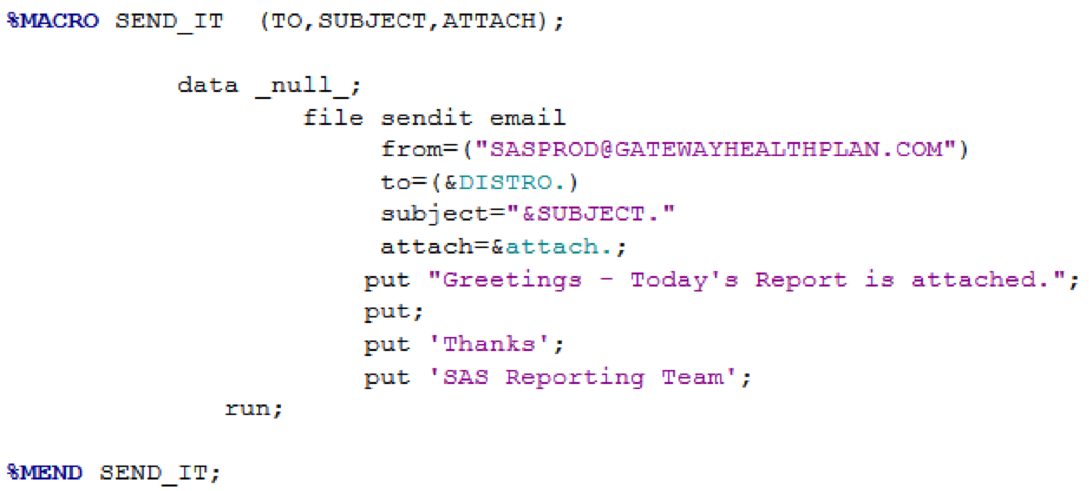
To □ Himes, Travis; □ Himes, Travis; □ Himes, Travis; □ Himes, Travis; □ Himes, Travis; □ Himes, Travis

Message | New_Customer_Report.XLSX (19 KB)

Greetings - The Daily New Customer Report is atatched.

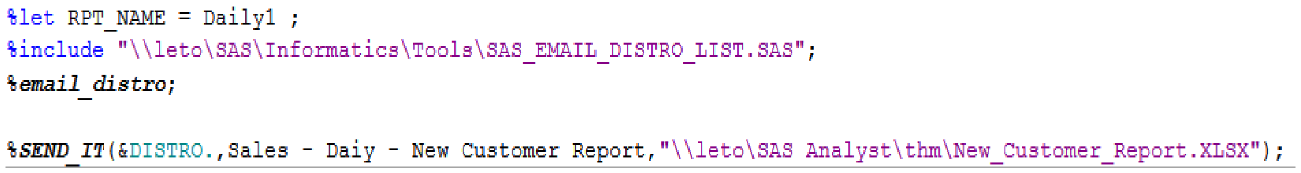
Thanks
SAS Reporting Team

The final recommendation to enhance SAS distributing reporting is modifying the above SAS email code into a macro called %SEND_IT which is to be saved outside of the SAS production code.



```
%MACRO SEND_IT  (TO,SUBJECT,ATTACH);

           data _null_;
                   file sendit email
                        from=("SASPROD@GATEWAYHEALTHPLAN.COM")
                        to=(&DISTRO.)
                        subject="&SUBJECT."
                        attach=&attach.;
                       put "Greetings - Today's Report is attached.";
                       put;
                       put 'Thanks';
                       put 'SAS Reporting Team';
              run;

%MEND SEND_IT;
```

This macro makes the subject and attachment of the email code dynamic. The standard SAS email code is a bit tedious so our suggestion is to have that code live outside of the production code. Therefore, we can use the following piece of code in all our SAS production codes. Within %SEND_IT the variable &TO will always be the value &DISTRO which is our reporting email lists of all users receiving the report. The variables &SUBJECT and &ATTACH can be modified by all SAS developers to meet the needs of each specific SAS report.



```
%let RPT_NAME = Daily1 ;
%include "\\leto\SAS\Informatics\Tools\SAS_EMAIL_DISTRO_LIST.SAS";
%email_distro;

%SEND_IT(&DISTRO.,Sales - Daiy - New Customer Report,"\\leto\SAS Analyst\thm\New_Customer_Report.XLSX");
```

The above code is without question more dynamic and more user friendly than the standard SAS Email Code we first discussed.

CONCLUSION

When creating a SAS production environment it is best practice to touch the SAS code that lives there as little as possible. While using the standard SAS Email Code will deliver your reports to the correct individuals it is often tedious to modify this code especially if it involves removing or adding users in large volume. Also, this approach does not involve keeping a SAS Distribution List which can be very beneficial when keeping a reporting inventory. Lastly, making changes to this SAS Distribution list is much easier than modifying the standard SAS Email Code. By creating a SAS Distribution list and SAS email macro which both live outside of your SAS production environment you are ensuring stability in the success rate of your production code and also saving yourself time when modifying distribution lists while keeping best practices.

ACKNOWLEDGEMENTS

We would just like to thank our BI & Analytics Team back in Pittsburgh. They provided much support throughout this process.

CONTACT INFORMATION

For any further questions feel free to reach out to us at:

Ryan Bixler:
Gateway Health Plan®
rbixler@gatewayhealthplan.com

Travis Himes:
Gateway Health Plan®
thimes@gatewayhealthplan.com