

Automating Distribution of Formatted Reports Using Base SAS®

Tracy DeCanio and John Blackwell, The Nature Conservancy, Arlington, VA

INTRODUCTION

Many industries have adopted the use of performance metrics to analyze their business processes and improve efficiency. However, without a full scale BI platform or the infrastructure to support SAS accounts for large numbers of clients, creating and distributing custom reports can be tedious and cumbersome. This paper details a simple, customizable method for large scale distribution of reports with individual, personalized data. Using the e-mail option in a *Data_NULL* step combined with a macro that creates the individual reports, you can easily e-mail a report to many people with their personal view of the data.

CREATING THE DISTRIBUTION LIST MACROS

The first step to this macro is creating your list of recipients. You will need, at the very least, some sort of unique identification field, such as an employee id. This field will be used to pull the data for that individual's report, and in this paper I have used a field called 'account_id'. The last piece you will need is the report recipient's e-mail address, which I have called 'emailaddress'. Once you have that dataset prepared, you will need to create a macro variable for each recipient and their e-mail address:

```
proc sql noprint;

    select count(*)
    into :NObs
    from recipients;

    select account_id
    into :account_id1-:account_id%left(&NObs)
    from recipients;

    select emailaddress
    into :emailaddress1-:emailaddress%left(&NObs)
    from recipients;

quit;
```

This statement will create a macro variable for each individual and their email address in the 'recipients' dataset. The first account_id will be stored as *&account_id1*, and the second will be stored as *&account_id2*. When the macro variables resolve, they will resolve to their actual value. For instance, if the first account_id is 'N1234', then *&account_id1* will resolve to 'N1234'. The *&NObs* macro will be used to tell the e-mail macro (below) how many times to iterate to ensure that each individual's report is created and sent.

CREATING THE REPORTS THROUGH A MACRO

First, start with the standard macro code to name your macro. Here I have named my macro 'email.' Second, you will need to have a *do loop* to cycle through your list of recipients. You will then need to pull the data for all of the fields of your report and format them as necessary for each individual.

```
%macro email;

%do i=1 %to &nobs;

    data report (keep=sales last_sale_date);
        set report_info_raw;
        format sales dollar10. last_sale_date worddate18.;
        sales=metric_one;
```

```
last_sale_date=metric_two;
if account_id= &&account_id&i;
run;
```

Note that in the first iteration of this macro (where $i=1$), the `&&account_id&i` macro variable referenced in the data step will first resolve to `&account_id1` and then resolve further to the value of the first `account_id`.

Next, you will need to store the reports for each individual to avoid overwriting the data set with the next person's data in the next iteration of the do loop. I have chosen to use a *PROC EXPORT* step to create an excel document that is uniquely named, but the file format can be anything you choose.

```
proc export
data=report
outfile="C:\report_location\report_&&account_id&i...xls"
dbms=dlm
replace;
delimiter='09'x;
run;
```

DISTRIBUTING THE REPORTS

Finally, you will need to e-mail the report to its owner. The filename statement with the email option allows you to send e-mail via a `data _null_` statement. There are numerous options for creating e-mail components, but I have chosen a few of the important ones below.

```
filename mymail email to=("&&emailaddress&i")
                        from= 'name@email.com'
                        replyto='name@email.com';

data _null_;
file mymail;
put '!em_subject!' 'Report';
put 'Hello,';
put 'Attached is your report.';
put '!em_attach!' "C:\report_location\report_&&account_id&i...xls ";
run;
```

The 'email' option in the filename statement will automatically use your default e-mail server on the local machine, and the default settings enable e-mail attachments. For more information on customizing the 'email' option in the filename statement, see the SAS online help for configuring system settings.

Close the do loop and end the macro:

```
%end;

%mend email;
```

Run the macro to create all of the reports that you will need to distribute and e-mail them to their recipients:

```
%email;
```

CONCLUSIONS

This method for creating and disseminating reports as laid out in this paper is a versatile and nimble way to satisfy

the constant appetite for data across a wide array of clients. While the application laid out here has been written under the guise of metrics reporting, there are clearly several other uses for such a process such as daily revenue updates for each operating unit in your company or daily audits of inventory and procurement. Using the simple macro described in this paper to create and e-mail individual reports to your data consumers, you can provide information with high impact and minimal effort.

REFERENCES:

Tilanus, Erik. Sending E-mail from the DATA Step. 038-2008

Zdeb, Mike. Creating Macro Variables Via Proc SQL.

ACKNOWLEDGMENTS

SAS is a Registered Trademark of the SAS Institute, Inc. of Cary, North Carolina.

CONTACT INFORMATION

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

Tracy DeCanio
The Nature Conservancy
4245 N. Fairfax Drive
Arlington, VA 22310
Work Phone: 703-841-4581
E-mail: tdecanio@tnc.org
Web: www.nature.org

John Blackwell
The Nature Conservancy
4245 N. Fairfax Drive
Arlington, VA 22310
Work Phone: 703-841-8786
E-mail: jblackwell@tnc.org
Web: www.nature.org

APPENDIX: FULL CODE WITH PRACTICE DATASETS

```
data report_info_raw;  
    input account_id $ metric_one metric_two mmddyy8.;  
    datalines;  
    N1234 500000 02-02-80  
    J2456 124000 01-07-09  
    C3879 423856 04-05-89  
    D5446 179000 02-06-09  
    ;  
run;
```

```
data employees;  
    length account_id $5. emailaddress $20. report $1.;  
    input account_id $ emailaddress $ report $;  
    datalines;
```

```

N1234 name_A@email.com Y
J2456 name_B@email.com N
C3879 name_C@email.com Y
P0987 name_D@email.com N
D5446 name_E@email.com Y
;
run;

```

```

data recipients (keep=account_id emailaddress);
    set employees;
    if report='Y';
run;

```

```

proc sql noprint;

```

```

    select count(*)
    into :NObs
    from recipients;

```

```

    select account_id
    into :account_id1-:account_id%left(&NObs)
    from recipients;

```

```

    select emailaddress
    into :emailaddress1-:emailaddress%left(&NObs)
    from recipients;

```

```

quit;

```

```

%macro email;

```

```

    %do i=1 %to &nobs;

```

```

    data report (keep=sales last_sale_date);
        set report_info_raw;
        format sales dollar10. last_sale_date worddate18.;
        sales=metric_one;
        last_sale_date=metric_two;
        if account_id= &&account_id&i;
    run;

```

```

proc export
    data=report
    outfile="C:\report_location\report_&&account_id&i...xls"
    dbms=dlm
    replace;
    delimiter='09'x;
run;

```

```
filename mymail email to=("&&emailaddress&i")  
                        from= 'name@email.com'  
                        replyto='name@email.com';
```

```
data _null_;  
    file mymail;  
    put '!em_subject!' 'Report';  
    put 'Hello,';  
    put 'Attached is your report.';  
    put '!em_attach!' "C:\report_location\report_&&account_id&i...xls ";  
    run;
```

```
%end;
```

```
%mend email;
```

```
%email;
```