Paper 078-2012

Sending E-mails in your sleep

Andy Hummel, Delta Air Lines, Atlanta, GA

ABSTRACT

In today's world of the mobile office it seems we are never totally disconnected from the workplace. An increase in global business has added to the need of always-available 24-hour a day communication. This often leads to an increase in working hours and a decrease in sleep. But SAS® can help by running automated programs and sending e-mails with dynamic data and attachments around-the-clock. SAS can also monitor processes and send e-mail alerts when a problem is detected. At Delta Air Lines SAS has increased efficiency through automation and has allowed programmers to send e-mails while they are literally at home sleeping.

INTRODUCTION

Ad hoc data requests often grow into regularly produced reports. Producing one or two simple reports a day can usually be done in a matter of minutes. However, as the complexity and number of reports increases SAS users can find themselves in report production and maintenance mode with little time for new development. Additionally, in the business of air transportation there is a need for seven day a week reporting which can create an inconvenience for the SAS programmer who enjoys having their weekends free.

This paper assumes the reader has a basic understanding of how to send e-mails using SAS. For a good overview of the fundamentals of sending e-mails you can refer to SUGI paper 178-29, "You've Got Mail – E-mailing Messages and Output Using SAS E-MAIL Engine" by Worden and Jones and SAS Global Forum paper 038-2008, "Sending E-mail from the DATA step" by Tilanus.

USING SAS TO SEND A BASIC E-MAIL WITH ATTACHMENTS

Delta uses daily reports to monitor numerous operational metrics such as the number of canceled flights, number of delayed flights and passenger load factors. These reports are produced in SAS and distributed in a PDF format through e-mail. In the early stages of report development, the SAS process that produced the reports were ran Monday through Friday by an analyst. As the value of the reports were realized, Saturday and Sunday reports were soon requested. The solution was to create scheduled SAS jobs and send the PDF reports as e-mail attachments. This freed the analyst from having to spend time sending reports on a daily basis and allowed for seven day a week reporting.

Below is an example of an e-mail with two PDF attachments sent from Andrew.Hummel@Delta.com to My.Boss@Delta.com and My.Other.Boss@Delta.com. The attached PDF reports are called "Daily_Cancelled_Flight_Report.pdf" and "Daily_Delayed_Flight_Report.pdf". The PDF reports are produced by SAS and stored on the SAS server.

To send an e-mail to multiple recipients simply list the addresses of the recipients in the TO statement. Likewise, multiple reports can be sent as attachments by listing the locations and names of the reports in the ATTACH statement.

```
FILENAME outbox EMAIL ("Andrew.Hummel@Delta.com");
DATA _NULL_;
       FILE outbox
       TO=("My.Boss@Delta.com" "My.Other.Boss@Delta.com")
       FROM=("Andrew.Hummel@Delta.com")
       SUBJECT=("Example of a SAS E-mail with Attachments" )
       ATTACH=("/sas_server/reports/Daily_Cancelled_Flight_Report.pdf"
               "/sas_server/reports/Daily_Delayed_Flight_Report.pdf");
       PUT " ";
       PUT "Hello Boss,";
       PUT " ";
       PUT "Attached are the Daily Operational Reports.";
       PUT " ";
       PUT "Andy Hummel";
RUN;
```

INSERTING DYNAMIC DATES, NAMES AND VALUES IN AN E-MAIL

By creating and passing variables into the subject and body areas of an e-mail the message can become more useful. For example, if we are measuring the number of delayed flights per day departing from MCO (Orlando, FL) we can create a variable that passes that number into the subject and body of the e-mail. Each morning when the MCO station manager checks his performance for the previous day he can quickly scan the subject line. If he reads that there were zero delayed flights he can simply delete the e-mail and move on. If the value is higher than zero he can open the attached report for more details.

This is especially useful if the end user receives the e-mail on a mobile device as it saves them from having to download the attachment and scan the report for data. By simply inserting the need-to-know summary data in the e-mail subject line and body, the e-mail can become more efficient and user friendly. The following sections of code will explain how to create macro variables in a data step and then pass the macro variables to the e-mail. For more information on macro variables and their use you can refer to SUGI 29 paper 243-29 “SAS ® Macro Programming for Beginners” by Slaughter and Delwiche.

First we will create the variables in a data step. The below code uses PROC SQL to sum the number of delays by date and station. The GROUP BY statement is essential in making sure the variables are summed at the date and station level.

```
PROC SQL;
CREATE TABLE count_of_delayed_flts_1 AS SELECT DISTINCT
       rpt_date,
       station,
       SUM(delays) as nbr_delays
FROM source_delayed
GROUP BY rpt_date, station
;QUIT;
```

The next step will take the date, station and number of delay variables from the data step and turn them into macro variables so they can be used in the e-mail. Macro variables are very powerful in that they can be used outside the data step and allow the placement of dynamic variables in an e-mail. The CALL SYMPUT routine creates a macro variable for each of our date, station and number of delays variables. We create the macro variable ‘RPT_DATE_FOR_E-MAIL’ from the DATA step variable ‘rpt_date’. Notice that ‘rpt_date’ and the other variables are placed inside the COMPRESS function. If COMPRESS is not used there could be blank spaces surrounding the macro variable. The COMPRESS function removes the blank spaces and makes the formatting cleaner.

```
DATA count_of_delayed_flts_2;
       SET count_of_delayed_flts_1;

       CALL SYMPUT('RPT_DATE_EMAIL',COMPRESS(rpt_date));
       CALL SYMPUT('STATION_EMAIL',COMPRESS(station));
       CALL SYMPUT('NBR_DELAYS_EMAIL',COMPRESS(nbr_delays));
run;
```

Finally the macro variables are used in the e-mail. We have created the macro variables 'RPT_DATE_EMAIL', 'STATION_EMAIL' and 'NBR_DELAYS_EMAIL'. The below e-mail example inserts the date, station name and number of late departure flights into the subject line and body of the e-mail. Notice that the & symbol is placed in front of the macro variable names. Also note that when you insert a macro variable in the PUT line, the text in the PUT line must be enclosed in double quotes. By creating and using macro variables the e-mail becomes dynamic and can change from day-to-day depending on the source data. This offers a great deal of flexibility and turns e-mails into powerful reporting tools.

```
FILENAME outbox EMAIL ("Andrew.Hummel@Delta.com");
DATA _NULL_;
       FILE outbox
       TO=("Andrew.Hummel@Delta.com")
       FROM=("Andrew.Hummel@Delta.com")
       SUBJECT=("There were &NBR_DELAYS_EMAIL Delays for &STATION_EMAIL on
       &DATE_EMAIL");
       PUT " ";
       PUT "Hello &STATION_FOR_EMAIL Manager,";
       PUT " ";
       PUT "On &DATE_EMAIL there were &NBR_DELAYS_EMAIL delays out of
       &STATION_EMAIL.";
       PUT " ";
       PUT "Andy Hummel";
RUN;
```

Below is the e-mail created by the SAS code above. As the e-mail shows the macro variables offer a great deal of flexibility of where to place the dynamic information. Since there is a limited amount of useful space in the subject line of the e-mail the most critical information should be placed here with the supporting information placed in the body. Obviously additional macro variables can be created and placed in the subject and body areas.

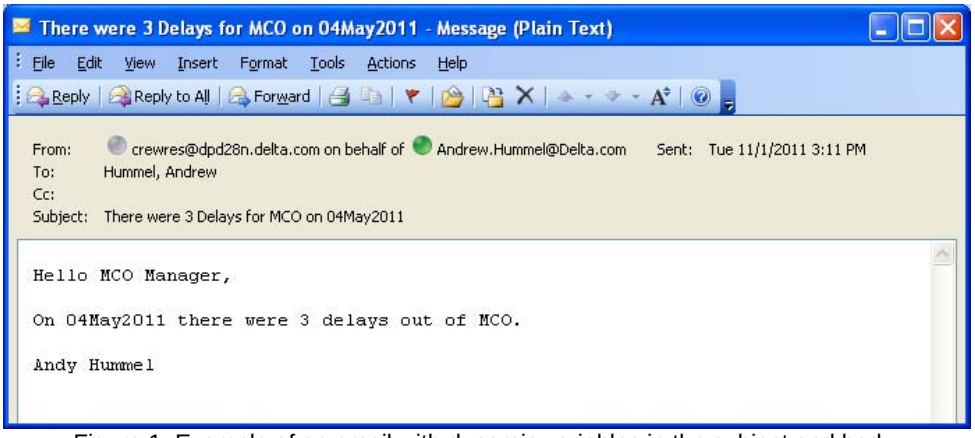

Figure 1. Example of an email with dynamic variables in the subject and body.

USING SAS TO MONITOR PROCESSES AND SEND ALERT E-MAILS

The source data for many of Delta's operational reports is stored in a Teradata data warehouse. The operational reports are scheduled to run early each morning and occasionally the source data for the reports has not completely loaded into Teradata when the reporting process starts. Sending out reports with partial data creates confusion and can cause the report readers to question the validity of the reports.

SAS can be used to check the source data prior to sending out the reports and if the source data is incomplete, SAS can stop the reporting process and send out an alerting e-mail. The following SAS code counts the number flights in Teradata for a given day, if the number of flights is below a defined parameter the process is aborted and an e-mail is sent to the SAS programmer informing them of the issue.

The below step counts the number of flights for a given day and creates a data step variable called 'nbr_flights'.

```
PROC SQL;
       CREATE TABLE flight_count_1 AS SELECT DISTINCT
              count(flight_numbers) as nbr_flights
       FROM source_flights
```

```
;QUIT;
```

The next step creates a macro variable called 'FLIGHT_COUNT' from the data step variable 'nbr_flights'. This variable is compared to an expected number, if the macro variable is below an expected number the SAS process is aborted and an e-mail is sent to the SAS programmer.

```
DATA flight_count_2;
       SET flight_count_1;
       CALL SYMPUT('FLIGHT_COUNT',nbr_flights);
run;
```

The below macro compares the number of flight numbers counted in Teradata to an expected amount of 1,000. If the number from Teradata is below the expected amount the SAS process is aborted and an e-mail to the SAS programmer is sent. In order to conduct the evaluation an %IF %THEN %DO statement is used inside a macro. Once the SAS process is aborted an alerting e-mail containing the number of observed records from Teradata is sent so that the programmer has an idea of how much data was loaded. The ABORT statement stops the SAS session so that incomplete reports are not sent.

```
%MACRO record_check;

       %IF &FLIGHT_COUNT. <1000 %THEN %DO;

       FILENAME outbox EMAIL ("Andrew.Hummel@Delta.com");
       DATA _NULL_;
       FILE outbox
       TO=("Andrew.Hummel@Delta.com")
       FROM=("Andrew.Hummel@Delta.com")
       SUBJECT=("!!! The Teradata Load is Incomplete !!!");
       PUT " ";
       PUT "The SAS reporting process has been stopped.";
       PUT "There were &FLIGHT_COUNT. records which is below the expected number of
1,000.";
       PUT "Please investigate.";
       PUT " ";
       PUT "Andy Hummel";

       ABORT RETURN;

       RUN;

       %END;

%MEND record_check;

%record_check();
```

Below is the resulting e-mail that is produced and sent by SAS letting the SAS programmer know that the process has been aborted and the number of records found in Teradata.

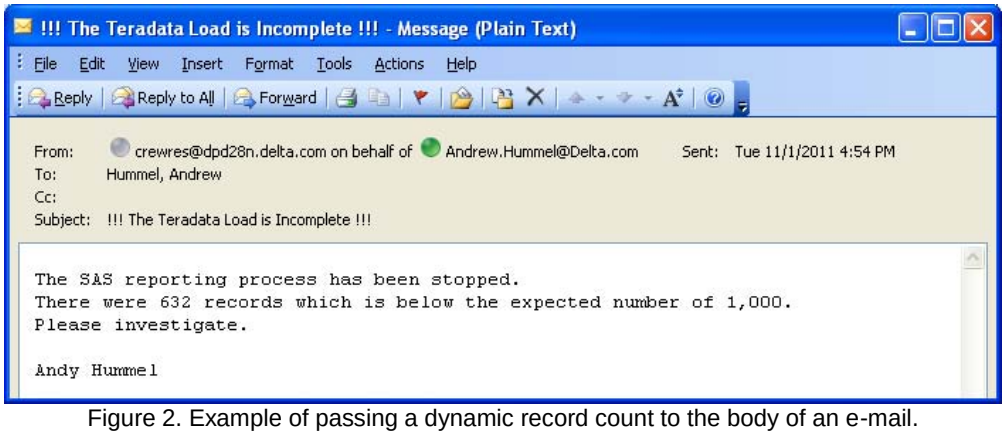

Figure 2. Example of passing a dynamic record count to the body of an e-mail.

The ability to evaluate variables and send alert e-mails has been applied to other applications at Delta. In the pilot training centers there are overhead monitors which display real-time information regarding training classrooms, hotel information and bus schedules for the pilots. The data is continuously updated and the source information is contained in a text file. The text file also contains a timestamp of the latest update. SAS is used to monitor the timestamp in the text file. Using the SLEEP function SAS imports the text file with the timestamp every 5 minutes and if the time of the last update is outside an expected range an e-mail is sent indicating that there is a problem with the data. A Word document attachment is also sent with instructions on how to restart the application along with the time of last update. This allows the issue to be quickly resolved often before the pilots notice that there is an issue. Additionally, the SAS job can run 24 hours a day without any human interaction.

Below is an example of the e-mail that is sent when the training data has been determined to be outside of the expected range.

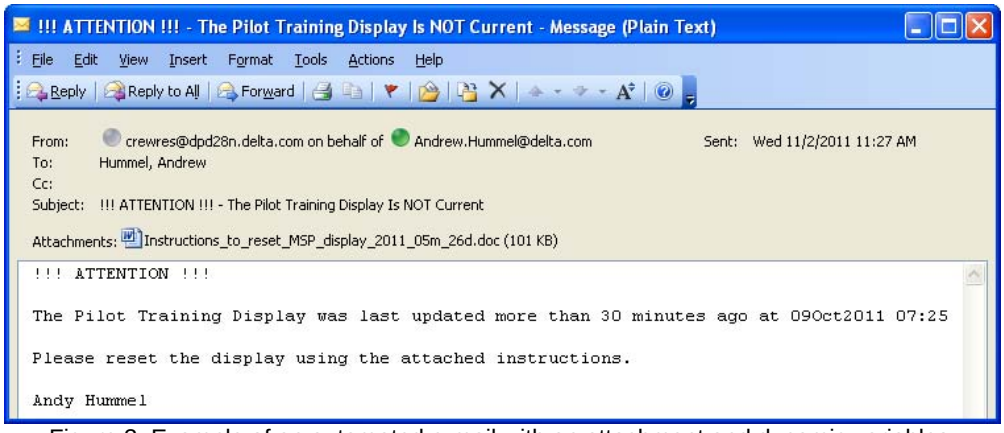

Figure 3. Example of an automated e-mail with an attachment and dynamic variables.

CONCLUSION

The above examples show how SAS programmers can literally create reports, monitor process and send e-mails in their sleep. SAS e-mails can be sent to multiple recipients with multiple attachments. Dynamic variables can be created and added to the subject and body areas of e-mail messages which allows users to quickly evaluate if they need to open the attachments. With some additional coding, dynamic variables can be compared to expected limits and if the variables are outside the limits altering e-mails can be sent and action taken prior to incorrect information being distributed. This all increases efficiency which allows the SAS programmer to spend more time sleeping, developing SAS programs or visiting Disney World.

CONTACT INFORMATION

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

Andy Hummel
Delta Air Lines
General Manager – Crew Resources
Andrew.Hummel@Delta.com
404-715-1270