

Paper 057-2011

Some Tricks to Get the Best Out of SAS® and UNIX

Chienchung Chen, HAVI Global Solutions, Downers Grove, IL

ABSTRACT

After years of using SAS under varied platforms, I enjoyed working under UNIX environment since it provided me with some easy-to-learn commands when I needed to perform varied development, production, and analytical tasks. All those commands, if applied appropriately, can improve the efficiency of the codes and the usage of resource. Besides, depending on the objectives of the tasks, the developer can choose between batch and interactive SAS mode and integrate the SAS codes with UNIX scripts to reduce the run time of the jobs and avoid errors. Furthermore, with the help of UNIX, the developer can even automate some tasks which usually require manual steps under PC environment. This paper will provide some tricks mentioned above.

INTRODUCTION

Basically through this paper, I will introduce some useful SAS procedures which work well under UNIX environment and then integrate them with UNIX scripts to automate the processes.

First, I will list the pros and cons of working under UNIX environment versus PC environment. Second, I will introduce ODS, for instance, ODS HTML, ODS PDF, etc. as alternatives to traditional DATA statement to help the developer export SAS dataset or results to varied formats. Furthermore, I will demonstrate the capabilities of ODS MARKUP statements (ODS TAGSETS.EXCELXP) which enables you to output multiple workbooks on one Excel file along with basic features like filter, freeze column, subtotal, etc. I include ODS MARKUP since it is one the most powerful and easy-to-learn statements I have ever used to generate reports. In fact, any of those procedures mentioned above can generate ready-to-use reports with appropriate formatting, not just a bunch of printed statistics. The developer can choose either of them to fulfill the tasks based on business needs.

In the next session, I will introduce basic UNIX CRON commands to help the developer to manage the submissions of the SAS jobs. Besides, the developer can also define the parameters which can be read as part of SAS inputs before the job is submitted. I will also introduce basic UNIX mailing commands. With the help of the mailing commands, the results can be automatically delivered to the final users, so the developer doesn't have to move the files among varied environments and manually send them out through Outlook.

BENEFITS OF USING UNIX VERSUS PC SAS

In general, UNIX and PC SAS have their own pros and cons depending on what your tasks are. I wrote a paper for Global Forum 2010 regarding the automation of using SAS DDE under PC. In this paper, I emphasized the advantages of utilizing VBA along with SAS DDE under PC environment to generate complex reports. However, if your task is simply to create rapid reports or to perform basic analyses and you don't have to generate any complicated graphics through SAS, UNIX SAS might be a good option for you. You can even utilize UNIX shell scripts to automate some of the tasks and improve the efficiency of the process, especially when you are dealing with huge datasets or multiple databases. Usually it is really time-consuming to perform joins or calculate basic statistics under PC SAS for huge datasets. Besides, you can control the launch time for your SAS jobs with basic UNIX commands and integrate the process with UNIX mailing scripts.

PROCEDURES WORK WELL UNDER UNIX ENVIRONMENT

In general, ODS CSVALL and ODS HTML work well under UNIX to generate basic CSV or EXCEL files. However, if you'd like to generate multiple reports and place them on different worksheets of the same Excel file, you might consider using ODS TAGSETS.EXCELXP. Besides, you can apply some EXCEL features like filter, subtotal, freeze column, etc. to the reports. In addition to those basic features, this procedure can also generate chart or pivot table without running SAS graph procedures.

Basic ODS Destination Syntax Structure

```
ODS LISTING CLOSE;
ODS CSVALL FILE = 'Myfolder/report.csv';
```

```
PROC statements;
```

```
RUN;
```

```
ODS CSVALL CLOSE;
ODS LISTING;
```

```
ODS LISTING CLOSE;
ODS HTML FILE = 'Myfolder/report.xls';
```

```
PROC statements;
```

```
RUN;
```

```
ODS HTML CLOSE;
ODS LISTING;
```

ODSVALL destination works extremely well when you don't care about the output layouts or formats, i.e. you only need to output the results. However, if your goal is to generate formatted reports, you should use ODS HTML/PDF/RTF destination and PROC TEMPLATE procedures.

Below I will demonstrate the basic structure and options of ODS TAGSETS.EXCELXP. This procedure is useful and UNIX friendly.

Sample ODS TAGSETS.EXCELXP

```
ODS LISTING CLOSE;
ODS TAGSETS.EXCELXP
PATH                = "&OUTLOC. /" /*output location*/
FILE                = "sample.xls"
STYLE               = Printer; /*define the user define or default styles*/

ODS TAGSETS.EXCELXP OPTIONS
(
SHEET_NAME          = 'Sheet1' /*name the worksheet for this first report*/
AUTOFIT_HEIGHT      = 'YES'
BANNER_COLOR_EVEN   = "white"
BANNER_COLOR_ODD    = "#eeeeee"
AUTOFILTER          = '1-3'
FROZEN_HEADERS      = '1'
FROZEN_ROWHEADERS   = '3'
EMBEDDED_TITLES     = 'Yes'
CENTER_VERTICAL     = 'Yes'
CENTER_HORIZONTAL   = 'Yes'
ABSOLUTE_COLUMN_WIDTH = '9, 30'
);

TITLE;
OPTIONS NONUMBER NODATE; /*first report*/
PROC PRINT DATA= report1 NOOBS LABEL;
RUN;

ODS TAGSETS.EXCELXP OPTIONS
(
SHEET_NAME          = 'Sheet2' /*name the worksheet for this second report*/
AUTOFIT_HEIGHT      = 'YES'
BANNER_COLOR_EVEN   = "white"
BANNER_COLOR_ODD    = "#eeeeee"
AUTOFILTER          = '1-3'
FROZEN_HEADERS      = '1'
FROZEN_ROWHEADERS   = '3'
EMBEDDED_TITLES     = 'Yes'
```

```

CENTER_VERTICAL          ='Yes'
CENTER_HORIZONTAL       ='Yes'
ABSOLUTE_COLUMN_WIDTH  ='9, 30
);

TITLE;

OPTIONS NONUMBER NODATE ;
PROC PRINT DATA= report2 NOOBS LABEL; /*second report*/
RUN;

ODS TAGSETS.EXCELXP CLOSE;
ODS LISTING;

```

Below I also list some useful options you can define while using EXCELXP destination. (Source from SAS Support Website http://support.sas.com/rnd/base/ods/odsmarkup/excelxp_help.html)

AUTOFILTER	Turn on auto filter for all columns or a range of columns
ABSOLUTE_COLUMN_WIDTH	List of widths to use for each column in a table no matter what
AUTO_SUBTOTALS	Add a subtotal function to the summary line of proc print
CENTER_VERTICAL	This value controls vertical centering for printing
CENTER_HORIZONTAL	This value controls horizontal centering for printing
EMBEDDED_TITLES	Put titles in the worksheet
FROZEN_HEADERS	Freeze rows from scrolling with the scrollbar
FROZEN_ROWHEADERS	Freeze columns from scrolling with the scrollbar
SHEET_NAME	Name the sheet of the sheet
HIDDEN_COLUMNS	range or list of column numbers to hide

In addition to those procedures mentioned above, there are also other ODS TAGSETS procedures which you can use for specific types of reports, for example, ODS TAGSETS.TABLEEDITOR. Those procedures are really powerful, but much more sophisticated and complicated than *EXCELXP destination*.

USEFUL UNIX COMMANDS

In this session, I will introduce UNIX CRON command and two mailing engines (MUTT and MAILX). With the help of those commands, we can automate our process and reduce the manual steps under UNIX.

Steps to Set Up CRON Jobs and Sample CRON Command

The purpose of CRON is to schedule the launch time for already-developed SAS jobs. So the SAS programs have to be written and debugged first before you put them on the CRON schedule.

SAS Program to be scheduled (program1.sas)

```
/sas9/sas /usr/home/myfolder/program1.sas
```

UNIX Script to run SAS job (/usr/home/myfolder/job1)

```
#!/usr/bin/csh
/sas9/sas /usr/home/myfolder/program1.sas
```

```
17 11 * * * /usr/home/myfolder/job1 05DEC08 01JAN09
02 * * * * /usr/home/myfolder/job2
```

UNIX

CRONTAB Command

```
crontab /usr/home/myfolder/cronfile
crontab -l
```

#1

#2

#3

#4

Steps

- After the SAS program is written and tested, create UNIX script file with/without parameters. (#1)
- List the script in your CRON profile file. The CRON profile needs be set to executable. (#2)
- Read this file into UNIX by using CRONTAB command. (#3)
- Verify if the process (#3) is completed successfully. (#4)
- Repeat the same steps when changes are made to the CRON profile. (#1- #4)

For details about how to schedule jobs, please refer to <http://adminschoice.com/crontab-quick-reference> .

```
* * * * * command to be executed
- - - - -
| | | | |
| | | | | +----- day of week (0 - 6) (Sunday=0)
| | | +----- month (1 - 12)
| | +----- day of month (1 - 31)
| +----- hour (0 - 23)
+----- min (0 - 59)
```

Besides, you can manage the CRON jobs by defining parameters in the CRON profile file. For example, program1 (shown above) is scheduled to run every day at 11:17 AM. Since parameter is defined, SAS will read "05DEC08 01JAN09" as parameter first before submitting the job. The method to read and interpret the parameter is shown below. You can use %SCAN to interpret the parameter. For example, &st will be given a token of '05DEC08' and &end, a token of '01JAN09'.

```
%let st=%scan(&sysparm,1);
%let end=%scan(&sysparm,2);
```

UNIX Mailing Commands

Another advantage of using UNIX is that you can integrate UNIX scripts to mail the reports or monitor the status of the jobs. Two of the most frequently used facilities are MIALX and MUTT mail Client. Depending on the tasks, you can choose between them. For SAS MACRO users, MUTT mail client might be a good choice when the task involves mailing reports to many recipients.

Sample UNIX MAILIT1 script

In addition to regular SAS mailing codes, you can utilize the basic UNIX scripts to deliver the messages or attachments. Below lists the basic UNIX mailing commands.

```

_FILENAME MAILIT1 PIPE "cat /usr/home/myfolder/mail.uue | MAILX -s ' SUBJECT LINE' jchen@havigs.com ";
X "UUENCODE /usr/home/myfolder/attachment.pdf attachment.pdf > /usr/home/myfolder/mail.uue";

DATA _NULL_;
FILE MAILIT1;
RUN;

X " rm /usr/home/myfolder/mail.uue";

```

Sample Mutt Mail Client Script

MUTT is a small but very powerful text-based mail client for UNIX operating.

For details, please refer to http://linux.about.com/library/cmd/blcmd11_mutt.htm. Below please find the basic mutt script.

```

.x "/opt/sfw/bin/mutt -a /usr/home/myfolder/report.pdf -s Report for JAMES CHEN ' jchen@havigs.com <
message.txt";

```

Integrate Mutt Mail Client with SAS Macro

For SAS Macro users, it is really easy to integrate MUTT mail client with SAS Macro to deliver multiple reports to varied recipients. Below you can find the single line MUTT script which will deliver reports to varied recipients according to mail_list dataset where recipients' IDs, first names, last names, and email addresses are saved.

```

%MACRO RUNMAIL (ID, FIRST_NAME, LAST_NAME, EMAIL);

```

```

x "/opt/sfw/bin/mutt -a /usr/home/myfolder/report.pdf -s Report for &FIRST_NAME &LAST_NAME ' &EMAIL < message.txt";

```

```

%MEND RUNMAIL;

```

```

%MACRO mailengine;

```

```

%LET inc=1;

```

```

%DO x=1 %TO &enum;

```

```

%IF &x=%EVAL((&inc*2)+1) %THEN %DO;

```

```

DATA _null_;

```

```

rc=SLEEP(30000);

```

```

RUN;

```

```

%LET inc=%eval(&inc+1);

```

```

%END;

```

```

DATA _null_;

```

```

SET mail_list;

```

```

IF ownerorder= &x ;

```

```

CALL SYMPUT("aa",ID);

```

```

CALL SYMPUT("bb",EMAIL);

```

```

CALL SYMPUT("cc",FIRST_NAME);

```

```

CALL SYMPUT("dd",LAST_NAME);

```

```

RUN;

```

```

%PUT &aa;

```

```

%RUNMAIL (&aa , &cc , &dd , &bb );

```

```

%END;

```

```

%MEND;

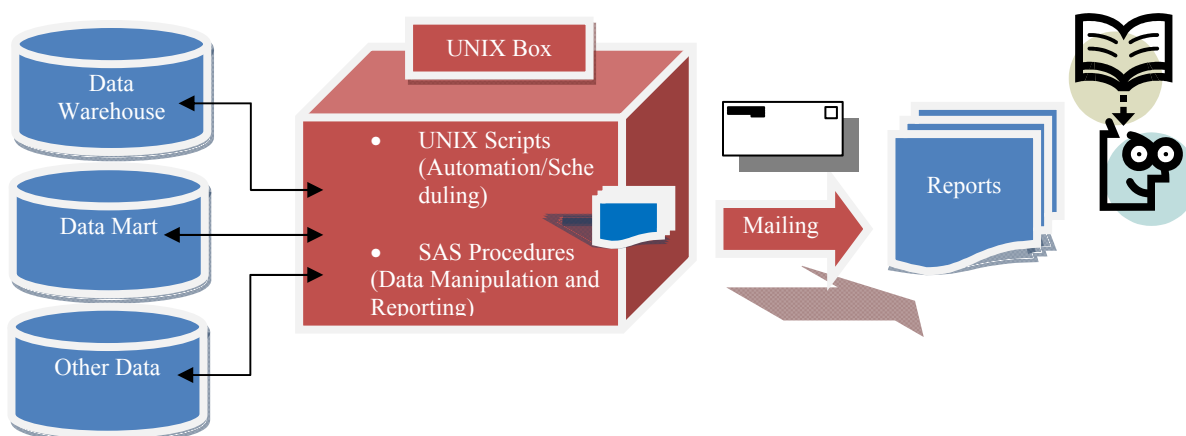
```

```

%mailengine;

```

CONCLUSION



Above graph demonstrates the basic concept of this paper. The goal of getting the best out of SAS and UNIX is to reduce manual steps, i.e. to automate the process as much as possible to save time and cost and improve the efficiency.

However, even there are lots of advantages of using SAS under UNIX, there are still some tasks which are difficult to achieve under UNIX. For example, if the reports involve with complicated graphics, it will be really time-consuming to develop the SAS codes under UNIX. In this case, PC SAS or Enterprise Guide might be better option. I wrote a proceeding for 2010 Global Forum to discuss the processes to generate complex reports using DDE and VBA. Please refer to <http://support.sas.com/resources/papers/proceedings10/074-2010.pdf> for details.

To summarize, my suggestion is that we should choose the best solutions available based on the tasks and accessible resources. Each platform has its own strengths and weakness. There is no need to stick to any platform as long as the single platform solution or 'hybrid' solution performs better.

REFERENCES

SAS Support Website http://support.sas.com/rnd/base/ods/odsmarkup/excelxp_help.html

The Beginners Guide to ODS MARKUP: Don't Panic! Eric Gebhart, SAS Institute Inc., Cary, NC (Paper 236-31 Available at <http://support.sas.com/rnd/base/ods/odsmarkup/p236-31.pdf>)

CONTACT INFORMATION

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

Chienchung Chen, Certified Advanced Programmer for SAS®9, MBA
 HAVI Global Solutions
 3075 Highland Parkway
 Suite # 200
 Downers Grove, IL 60516
 Phone 630-493-6654
 Email jchen@havigs.com
 Web www.havigs.com



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.