

Retention

You and your colleagues work very hard to create stored processes and deliver them to various departments in your organization to review the reports on regular basis and on time. Have you ever wondered how many of the reports you provide to the audience are actually being used? This paper presents a neat way to identify who ran the stored processes and what time they have run the stored processes by scanning all the stored process server log files and generates a list report and a frequency report.

Reporting Stored Process Server Log Files

First step is to identify where the stored process log files are stored. You may have to talk to your SAS® Administrator in your organization and get the path and access to these files. It is most likely that these files are stored in the following location <http://publibprod.fw.ibm.com/pubservlet.ibmredbooks.ibm.com/rdbooks/open.do?number=SG24522> and the usually named as SASAP_STPServer_YYYYMMDD_XXXX. There are many ways to get the list of file names from a folder and store it in a dataset. I am using the following Figure 1.

```
Figure 1.
macro get_filenames(location);
filename _dir_ "%qstr(%location,);";
data filenames (keep=filename where=(index(upcase(memname),'SASAP_STPSERV') > 0));
bandersopenp ' dir_ ' ;
if handle > 0 then do;
  count=round(handle);
do _i=1 to count;
  memname=reads(handle,1);
  output
  filenames;
end;
end;
end;
run;
filename _dir_ clear;
%get_filenames(your stored process server log files path)
```

Creating Macro variables

Next, we need to know how many log files we have to import, this can be done using SQL procedure by counting number of log files and storing in a Macro variable &count and also of log files in a macro variable &stplogs using into: in SQL procedure. Both the macro variables will be used in subsequent steps.

```
Figure 2.
proc sql noprint ;
select count (distinct memname )
into: &count from filenames
where
input(scan(memname,3,'_'),ANYDTYDEILL.) < today();
select distinct memname
into: &stplogs separated by ' ' from filenames
where
input(scan(memname,3,'_'),ANYDTYDEILL.) < today();
quit;
```

Reporting Stored Process Server Log Files

Here is the sample stored process server log file.

```
Figure 3.
Sample line from Figure 3.
2012-08-27 09:45:42 288 INFO 0000X060 B bhargav.acharya@REATA - STP: 3: Executing E:\sasor4\4002903\Programs\Stored Process Test1.sas.
This is basically telling us that a stored process Test1.sas is executed and the user, date and time (Bhargav.Acharya_2012-08-27_9:45:42) who submitted the requests to the stored process server.
```

```
Figure 4.
%macro doelogs();
  %do i=1 %to &count;
  %let ds=%scan(&stplogs,&i,|str|);
  DATA logfil (where=(index(upcase (&ts),'EXECUTING') > 0));
  infile "&path of your stored process server log files\&ds";
  input txt $1000.;
  %let;
  %mend;
  %doelogs;
%run;
```

Reporting

All we need from this final data set is user name, date and time of when the stored process was run and the name of the stored process. We can employ various SAS® functions to extract the information we need. I am not pasting the code how I got to the report data set.

```
Figure 6.
DATA report;
set final;
/SAS® Statements and functions you can think of to extract User,Date and Time of Stored process run from all the log files will go here;
label
  Datein    = "Date of Run"
  Time      = "Time of Run"
  loc_up    = "Location and Stored Process"
  location  = "Location of the Stored Process"
  stprocess = "Stored Process"
  ;
run;
libname ext 'Output Library path';
proc sort data = report nodupkey out=ext.report;
  by user datein time stprocess;
run;
```

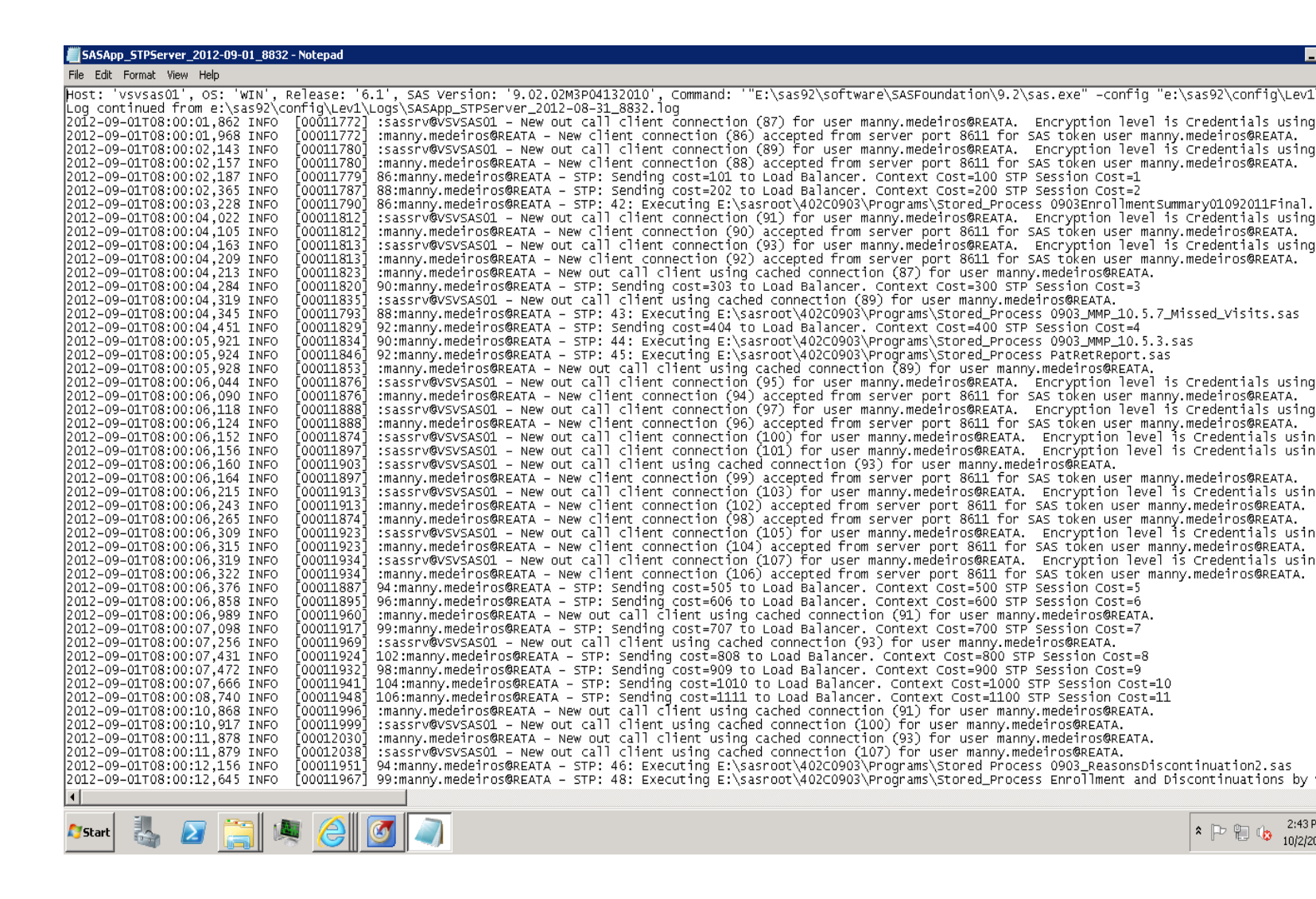
Once we have the data set with all the necessary information that we need in front of us, we will now use the basic SAS® procedures like PRINT procedure and FREQ procedure to present a list and a frequency reports.

```
Figure 7.
title 'List Report.';
proc print data=ext.report label;
var user stprocess location Datein Time;
run;

title;
title 'Frequency Report by User.';
proc freq data=ext.report ;
  by user;
  table stprocess / noout noprint;
run;
```

Here are the sample outputs of list and frequency reports.

Figure 8.



Concatenation and Cleaning

We now have one data set per one log file. The task ahead of us is to stack all the data sets on the top of each other and deleting the data sets that we won't be using from our work library. To accomplish this task we will loop through the data step &count times (number of log files times) to create a final data set by concatenating all the individual logfil, logfil, logfil data sets and use DATASETS procedure to delete the data sets that are not required from the work library.

```
Figure 5.
%macro combine;
DATA final;
set
&do 1 = 1 %to &count;
logfil;
%do i=1 %to &count;
proc datasets lib=work nolist;
delete logfil;
%let;
%end;
%combine;
run;
```

Reporting Stored Process Server Log Files

All we need from this final data set is user name, date and time of when the stored process was run and the name of the stored process. We can employ various SAS® functions to extract the information we need. I am not pasting the code how I got to the report data set.

```
Figure 6.
DATA report;
set final;
/SAS® Statements and functions you can think of to extract User,Date and Time of Stored process run from all the log files will go here;
label
  Datein    = "Date of Run"
  Time      = "Time of Run"
  loc_up    = "Location and Stored Process"
  location  = "Location of the Stored Process"
  stprocess = "Stored Process"
  ;
run;
libname ext 'Output Library path';
proc sort data = report nodupkey out=ext.report;
  by user datein time stprocess;
run;
```

Reporting

Once we have the data set with information of user names, date and times of the when and what stored process is run, we could use it to send emails based on certain rules, for example if you want a certain group of people in your country to review certain reports n number of times, you can send them reminders via emails. For a good overview of the fundamentals of sending e-mails you can refer to SAS® 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 Tianus and SAS® Global Forum 2012 paper 078-2012, "Sending E-mails in your sleep" by Andy Hummel.

Stored processes with very few hits or no hits across the users could be a sign that users may have not understood the reports and its use. You may want to re-visit the stored process, educate how the report is designed and how it ought to be used.

Retention

With SAS® DATA step and little bit of macro programming you could keep an eye on your stored processes and who and when those stored processes are being accessed by users within your organization and you can take it a bit forward by sending emails to the user who are not reviewing the reports in a timely manner.

References

1. SAS 9.1® Macro Language Reference.
2. SAS Stored Processes 2.0 Developer's Guide.
3. SAS Enterprise Guide 4.2®.
4. <http://stackoverflow.com/questions/140543using-sas-macro-to-join-a-lot-of-filenames-from-a-windows-directory>.

Contact Information

Your comments and questions are valued and encouraged. Contact the author at:
Name: Bhargav Acharya
Phone: 408-751-5558
E-mail: acharya.bhargav@gmail.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.