Paper 161-2017

Tracking SAS Licensed product usage.

Victor L. Andruskevitch and Ankur Patel

ABSTRACT

This paper offers a methodology for tracking who is using your licensed SAS products and which products are being used. Knowing how often each product is used helps your organization determine the number of SAS product licenses needed. The source code included in this paper is Linux specific but can easily be modified to run on the Windows platform.

INTRODUCTION

SAS has provided us with three utilities which when used together help us determine if and when our SAS licensed products are used. The first utility is the sas.tools.viewregistry.jar. This utility is intended to determine what products are installed at your location and what hotfixes are needed. As a result of executing this utility you will find a file named DeploymentRegistry.html. Within that file are entries for every product installed on your servers. Those entries resemble the following.

Host: lax
Product Code: tmine
Version: 14.1
Display Name: SAS Text Miner
Display Version: 14.1

From this entry the fields of interest are “Product Code” and “Display Name”. We use this information later when we provide the program code to record our product usage. The second utility of interest is RTRACE. We want to enable this utility in our workspace and stored process server sasv9_usermods.cfg files. Adding the options to turn on RTRACE logging which produces log files for every file accessed by your users program during execution. Those options should be as follows**.**

-RTRACE ALL

-RTRACELOC “/path to your log storage location/PID%p_DATE%d:%t.log”

I choose to encode the process identifier along with the log creation date and time in the filename. I do this because Linux does not record file creation datetime. I recommend saving these logs to a separate directory. The third SAS provided utility we use is named “tslvl”. This utility is used in our SAS program to decode the “Product Code” for every SAS executable file accessed by your users programs.

Now that you have become acquainted with the three utilities used, it’s time to take a look at the format of the RTRACE logs produced. Below is an excerpt from one of our logs.

```
File opened: /SASHome/SASAppPreDev/SASFoundation/9.4/sasexe/tkexml.so
File opened: /SASHome/SASAppPreDev/SASFoundation/9.4/sasexe/t0a8en.so
File opened: /SASHome/SASAppPreDev/SASFoundation/9.4/sasexe/tk4aroll.so
File opened: /SASHome/SASAppPreDev/SASFoundation/9.4/sasexe/tk4aunxf.so
File opened: /SASHome/SASAppPreDev/SASFoundation/9.4/sasexe/tk4aslog.so
File opened: /SASHome/SASAppPreDev/SASFoundation/9.4/sasexe/tkperf.so
```

At this time it helps to examine the program used to parse these logs into datasets. I’ve always found it easier to follow along reviewing the comments in code to understand what a program is doing. I hope this method works well for everyone:

```
/*****************************************************************************/
/* Program: PARSE_RTRACE_LOGS.sas                                            */
/*                                                                           */
/* Purpose: This program retrieves the file attributes owner, modified       */
/*          along with the creation timestamp and process id from the        */
/*          filename.                                                        */
/*****************************************************************************/
options msglevel=i;
/* location for historical dataset                                           */
libname in1 "/SASHome/{dataset location}/rtrace";
/* location where rtrace logs are stored                                     */
%let subdir=/SASHome/{log location}/Logs/;

data work.dirlist(keep=filename strUser ModDT CrDT strPID strSource);
/* must be explicitly set to blank to have the filename function generate a fileref    */
 fileref = '        ';
 rc = filename(fileref,"&subdir");
/* open fileref and loop throughfilenames                                    */
did=dopen(fileref);
do i = 1 to dnum(did);
/* extract filename                                                          */
   fileName = dread(did,i);
/* find the position of PID in the filename                                  */
   pid_start = index(fileName,"PID")+3;
/* find the end position of PIF in filename                                  */
   pid_end =  index(fileName,"DATE")-1;
/* extract PID                                                               */
   strPID = substr(fileName,pid_start,pid_end - pid_start);
/* find the start of DATE in filename                                        */
   date_start = index(fileName,"DATE")+4;
/* find the end position of DATE in the filename                             */
   date_end =  index(fileName,":")-1;
/* set the start of the time field                                           */
   time_start = date_end +2;
/* find the end of the time field                                            */
   time_end = index(filename,".");
/* extract creation time                                                     */
   tmpTime = Trim(substr(filename, time_start, time_end - time_start));
   tmpHour = Substr(tmpTime,1, 2)*1;
   tmpMin = Substr(tmpTime,3,2)*1;
   tmpSec = Substr(tmpTime,5,2)*1;
/* extract creation date                                                     */
   strDT = trim(substr(fileName,date_start,date_end - (date_start-1)));
   tmpDT = input(put(strDT,8.),yymmdd8.);
   CrDt = DHMS(put(tmpDT,8.),tmpHour,tmpMin,tmpSec);
   fid = mopen(did,fileName);
   infonum=foptnum(fid);
   if fid then do;
/* extract userid                                                            */
      strUser=finfo(fid, foptname(fid, 2));
/* extract file modified date                                                */
      ModDT=input(trim(finfo(fid, foptname(fid, 5))),datetime20.);
      format ModDT CrDT datetime20.;
/* extract file location                                                     */
      strSource = finfo(fid, foptname(fid, 1));
      output;
   end;
end;
rc = dclose(did);
rc = filename(fileref);
run;

/* read the contents of each file                                            */
data work.new(keep=filename strUser ModDT CrDT strPID strSource strAction strMod strExe);
```

```
  length strSource $110;
  length name $25;
  set work.dirlist;
  filepath = "&subdir"||fileName;
  infile dummy filevar = filepath truncover end=done;
  do while (not done);
    strSource = filepath;
    input rec $110.
       strAction $ 6;
       ExeStart = index(rec,'/sasexe/');
/* we are ignoring all but executable files                                  */
       if ExeStart GT 0 then do;
         Len = Lengthc(Trim(rec));
/* extract executable filename                                               */
         strExe = substr(rec,ExeStart+8, (Len - (ExeStart + 8))+1);
/* KLUDGE TO PREVENT REPORTING SAS INITALIZATION OF INSTALLED PRODUCTS       */
/* YOUR INSTALLED PRODUCTS MAY DIFFER                                        */
         select (strExe);
           when ('uwyl0001') strExe='';
           when ('uwyl0003') strExe='';
           when ('uwyl0005') strExe='';
           when ('uwyl0007') strExe='';
           when ('uwyl0008') strExe='';
           when ('sasxglob') strExe='';
           when ('sasxgopt') strExe='';
           when ('sasnk002') strExe='';
           when ('sasnk003') strExe='';
           when ('sasnk004') strExe='';
           when ('sasnk006') strExe='';
           when ('sasnk007') strExe='';
           when ('sasnk009') strExe='';
           when ('sasnk010') strExe='';
           when ('sasnk100') strExe='';
              otherwise;
         end;
/* return software code                                                      */
         if strExe NE "" then do;
           strMod = tslvl(strExe, 'p');
           if strMod NE "" then do;
         output;
           end;
         end;
       end;
  end;
run;
/* update product name based on software code                                */
data by_product(keep=strSource strAction strPID strMod strProdName strUser ModDT CrDT
strExe);
format strProdName $120.;
  set new;
    select (strMod);
/* you may need to add or remove your installed products from this select when block  */
       when ('_no128b_') strProdName='SAS Enterprise Miner Server';
       when ('acceldb2fmt') strProdName='SAS Formats Library for DB2';
       when ('accelmva') strProdName='DATA Step to DS2 Translator';
       when ('accelmvadb2') strProdName='SAS Accelerator Publishing Agent for DB2';
       when ('accelmvahadp') strProdName='SAS Accelerator Publishing Agent for Hadoop';
       when ('accelmvanetz') strProdName='SAS Accelerator Publishing Agent for Netezza';
       when ('accelmvaorcl') strProdName='SAS Accelerator Publishing Agent for Oracle';
       when ('accelnetzfmt') strProdName='SAS Formats Library for Netezza';
       when ('accesssample') strProdName='SAS/ACCESS Interface Products Samples';
       when ('base') strProdName='Base SAS';
       when ('connect') strProdName='SAS/CONNECT';
```

```
        when ('db2') strProdName='SAS/ACCESS Interface to DB2';
        when ('dcmcomdsvrc') strProdName='SAS Decision Manager Common Data Server';
        when ('dcmsvr') strProdName='SAS Enterprise Decision Management Common Server';
        when ('dmine') strProdName='SAS Enterprise Miner';
        when ('dminethin') strProdName='SAS Enterprise Miner Client';
        when ('dmscore') strProdName='SAS Data Mining Scoring';
        when ('dquality') strProdName='SAS Data Quality Server';
        when ('dtdrodbcmssq') strProdName='DataDirect ODBC Driver for Microsoft SQL Server';
        when ('eguide') strProdName='SAS Enterprise Guide';
        when ('ets') strProdName='SAS/ETS';
        when ('graph') strProdName='SAS/GRAPH';
        when ('iml') strProdName='SAS/IML';
        when ('or') strProdName='SAS/OR';
        when ('qc') strProdName='SAS/QC';
        when ('stat') strProdName='SAS/STAT';
        when ('etscomp') strProdName='SAS/ETS Time Series & Econometric Modeling Common
Components';
        when ('gridnode') strProdName='SAS Grid Manager';
        when ('gridsasgsub') strProdName='SAS Grid Manager Client Utility';
        when ('hadoop') strProdName='SAS/ACCESS Interface to Hadoop';
        when ('hadoopbasics') strProdName='BASE Infrastructure to support Hadoop';
        when ('hadoopwrapr') strProdName='DataFlux API for Hadoop';
        when ('hdoopsasjars') strProdName='SAS/ACCESS to Hadoop JAR Files';
        when ('hps') strProdName='SAS High-Performance Server';
        when ('inttech') strProdName='SAS Integration Technologies';
        when ('intechsrv') strProdName='SAS Integration Technologies';
        when ('mmcommon') strProdName='SAS Model Manager Common Components';
        when ('mssqlserver') strProdName='SAS/ACCESS Interface to Microsoft SQL Server';
        when ('netezza') strProdName='SAS/ACCESS Interface to Netezza';
        when ('netessabndl') strProdName='SAS/ACCESS Interface to Netezza';
        when ('odbc') strProdName='SAS/ACCESS Interface to ODBC';
        when ('odstemplate') strProdName='ODS Templates';
        when ('oledb') strProdName='SAS/ACCESS TO OLE DB';
        when ('oracle') strProdName='SAS/ACCESS Interface to Oracle';
        when ('pcfile') strProdName='SAS/ACCESS Interface to PC Files';
        when ('platfmltypes') strProdName='SAS Intelligence Platform Object Framework';
        when ('platformlsf') strProdName='Platform LSF';
        when ('securedom') strProdName='SAS/SECURE';
        when ('securessl') strProdName='SAS/Secure SSL';
        when ('share') strProdName='SAS/SHARE';
        when ('sybase') strProdName='SAS/ACCESS SYBASE SQL';
        when ('spds') strProdName='SAS Scalable Performance Data Server';
        when ('spdsclient') strProdName='SAS Scalable Performance Data Client';
        when ('studio') strProdName='SAS Studio Basic';
        when ('sybase') strProdName='SAS/ACCESS Interface to Sybase';
        when ('tablesrvtk') strProdName='SAS Table Server Base Components';
        when ('tmine') strProdName='SAS Text Miner';
        when ('txtancomp') strProdName='SAS Text Analytics Common Components';
        when ('txtandocconv') strProdName='SAS Document Conversion';
        otherwise strProdName='NULL';
    end;
  if strProdName NE 'NULL' then do;
    output;
  end;
run;
/*summarize dataset                                                              */
proc sql;
create table history as
SELECT distinct by_product.strUser, by_product.strPID, strMod, strProdName,
min(by_product.CrDT) format=datetime26. AS Start, max(by_product.ModDT) format=datetime26. AS
Finish
FROM by_product
GROUP BY by_product.strUser, by_product.strPID;
```

```
/* append to historical dataset                                                    */
proc append base=in1.history data=history;run;
proc sort data=in1.history nodup out=in1.history;
  by strUser strPID Start Finish;
run;
```

Now that we have a historical dataset we can report on it as we see fit. This example should be moved to a separate file and the previous code scheduled to run on daily as close to midnight as possible to capture daily activity:

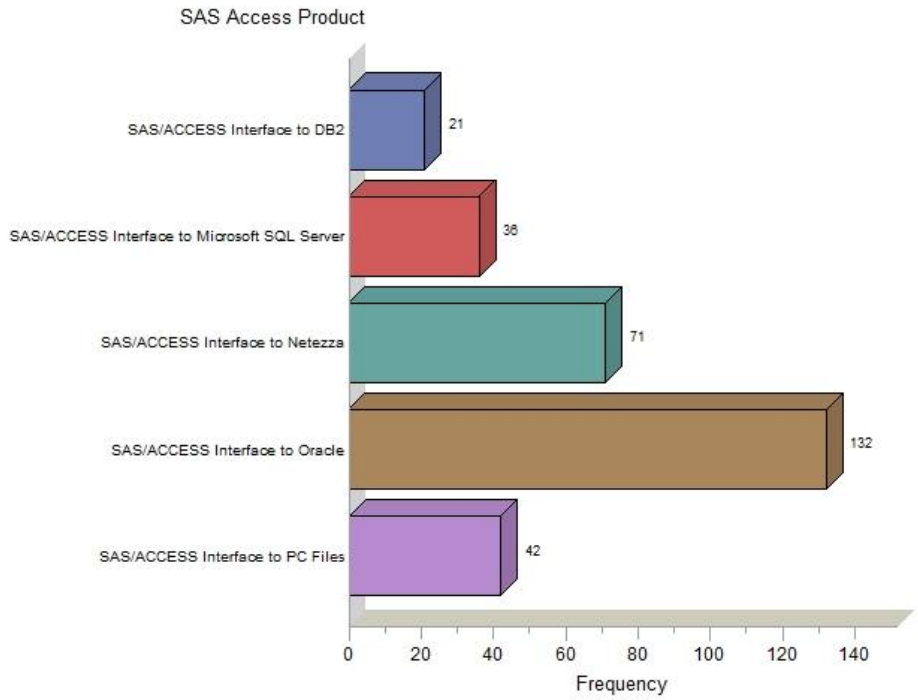
```
data _null_;
  call symput('rptto',trim(left(put(today(), worddate.))));
  call symput('rptfrom',trim(left(put(intnx('day',today(),-30), worddate.))));
run;
*****************************************************************;
* Create a report of SAS/ACCESS License Usage by Date/Product     *;
*****************************************************************;
proc sql;
create table access_use as
select distinct strProdName, datepart(Start) format date9. as DP, Count(strMod) as MOD
from rtrace.history
where datepart(Start) between today() and intnx('day',today(),-30) and strProdName like
'SAS/ACCESS%'
group by DP, strProdName
order by DP, strProdName;

proc print noobs label data=access_use;
  by DP;
label DP='Date';
label strProdName= 'Product Name';
label MOD= 'SAS Access Product frequency';
var strProdName;
sum MOD;
title1 'Number of Times SAS Product was used';
title2 'Summarized by Date';
title3 "For &rptfrom through &rptto";
footnote;
run;

proc sql;
create table access_use_graph as
select strProdName
from rtrace.history
where strProdName like 'SAS/ACCESS%';
Axis1
       STYLE=1
       WIDTH=1
       LABEL=( "SAS Access Product" );
Axis2
       STYLE=1
       WIDTH=1
       MINOR=
       (NUMBER=1);;
TITLE;
TITLE1 "SAS Product Usage";
PROC GCHART DATA=access_use_graph;
       HBAR3D
        strProdName
 /     SHAPE=BLOCK
FRAME  TYPE=FREQ
FREQ
       LEGEND=LEGEND1
       COUTLINE=BLACK
```

```
      MAXIS=AXIS1
      RAXIS=AXIS2
PATTERNID=MIDPOINT;
RUN; QUIT;
```

Graph 1, SAS Access Product Usage across Grid Servers.



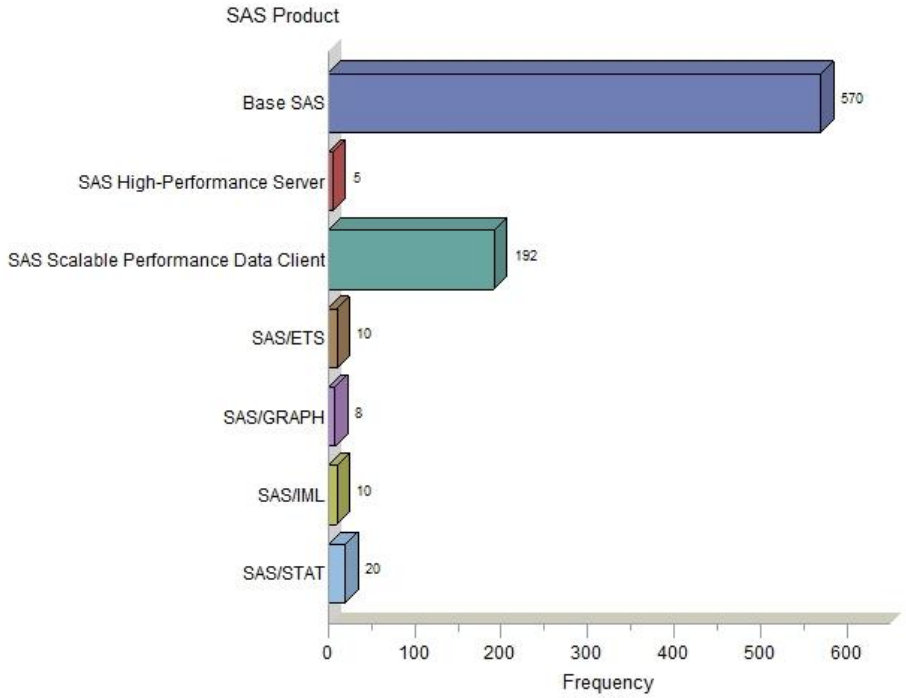
We are licensed for a number of SAS Access engines. These metrics validate our company's investment. Additionally we can show which users are invoking the Access engines.

```
*****************************************************************;
* Create a report of SAS Usage by Date/Product                  *;
*****************************************************************;
proc sql;
create table prod_use as
select distinct strProdName, datepart(Start) format date9. as DP, Count(strMod) as MOD
from rtrace.history
where datepart(Start) between today() and intnx('day',today(),-30) and strProdName not like
'SAS/ACCESS%'
group by DP, strProdName
order by DP, strProdName;

proc print noobs label data=prod_use;
  by DP;
label DP='Date';
label strProdName= 'Product Name';
label MOD= 'SAS Product frequency';
var strProdName;
sum MOD;
title1 'Number of Time SAS Product was used';
title2 'Summarized by Date';
title3 "For &rptfrom through &rptto";
footnote;
run;
```

```
proc sql;
create table prod_use_graph as
select strProdName
from rtrace.history
where strProdName not like 'SAS/ACCESS%';
Axis1
       STYLE=1
       WIDTH=1
       LABEL=( "SAS Product" );
Axis2
       STYLE=1
       WIDTH=1
       MINOR=
       (NUMBER=1);;
TITLE;
TITLE1 "SAS Product Usage";
PROC GCHART DATA=prod_use_graph;
       HBAR3D
        strProdName
 /     SHAPE=BLOCK
FRAME  TYPE=FREQ
FREQ
       LEGEND=LEGEND1
       COUTLINE=BLACK
       MAXIS=AXIS1
       RAXIS=AXIS2
PATTERNID=MIDPOINT;
RUN; QUIT;
```

Graph 2, SAS Product Usage across Grid Servers.



Knowing which products are being used and how frequently is information your manager can use to insure that the proper mix of products is licensed.

CONCLUSION

The methodology I have outlined should afford you the opportunity to audit your organizations SAS product usage. This will enable you to tell your manager what products are being used and if necessary by who. You may find that your company has licensed products that are seldom used or you may need additional licenses based upon usage. I hope you find this program useful.

Caution: The datetime stamps do not reflect actual SAS product usage duration, they only indicate user's session duration.

REFERENCES

Filelist macro., Don Henderson. Available: http://www.sascommunity.org/wiki/Filelist_macro

Measuring SAS Software Usage on Shared Servers with the RTRACE Facility. , Raithel, Michael A. Proceedings of the 29th Annual SAS Users Group International Conference.
Available: http://www2.sas.com/proceedings/sugi29/215-29.pdf

ACKNOWLEDGMENTS

I would like to thank Todd Burkard, from Medical Health Network, Michael Raithel, from Westat and Rajeev Gagneja for their critique and suggestions.

CONTACT INFORMATION

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

Victor Andruskevitch
Bessler Consulting & Research
LeRoy_Bessler@wi.rr.com

DISCLAIMER

The contents of this paper are the work of the authors. Use of the included program code is without warranty and is provided "as is".