

Supporting a SAS® Server Architecture

Angela Hall, Zencos Consulting, Cary, NC

ABSTRACT

Your organization has purchased a SAS® Solution, which could be for Business Intelligence, Warranty Analysis, Anti Money Laundering or any number of other offerings. Sitting in a server farm the hardware is maintained but you are given the responsibility of managing the SAS solution environment. How do you organize the administration work and keep yourself out of the support game? This paper will provide a framework for planning SAS server administration.

DOCUMENT ROUTINE SAS SYSTEM TASKS

SAS Installation team will provide information on what was installed and how services can be managed. Pull out the information and place in an internal SAS knowledge base so multiple people can quickly find and respond to incoming tickets.

1. How to Stop and Restart SAS Services

SAS Support provides general instructions in Online Documentation:

<http://support.sas.com/onlinedoc/913/getDoc/en/bisag.hlp/a003133688.htm>

In a windows environment, set up a batch script with net stop and net start commands, similar to the code below:

```
net stop "SAS - Tomcat 4.1.18"
net stop "SAS Remote Services"
net stop "SAS Lev1 SS - 9.1"
net stop "SAS Lev1 OB - 9.1"
net stop "SAS Metadata Server 9.1"
net stop "Xythos"

:# Sleep to ensure Services Stop Successfully and files become unlocked
"C:\Program Files\Windows Resource Kits\Tools\sleep.exe" 60

net start "Xythos"
net start "SAS Metadata Server 9.1"
net start "SAS Lev1 OB - 9.1"
net start "SAS Lev1 SS - 9.1"
net start "SAS Remote Services"
net start "SAS - Tomcat 4.1.18"
```

With multiple servers, the ability to stop and start all servers associated SAS services in the proper order and in one script is ideal. Microsoft offers a PSSERVICE tool to easily do this via command lines. <http://technet.microsoft.com/en-us/sysinternals/bb897542.aspx>

For remote services, all that is needed is the following change to a net stop/start line such as those above. Remove the 'net' and add 'psservice \machine-name'

```
psservice \machine-name stop "SAS Analytics Platform"
```

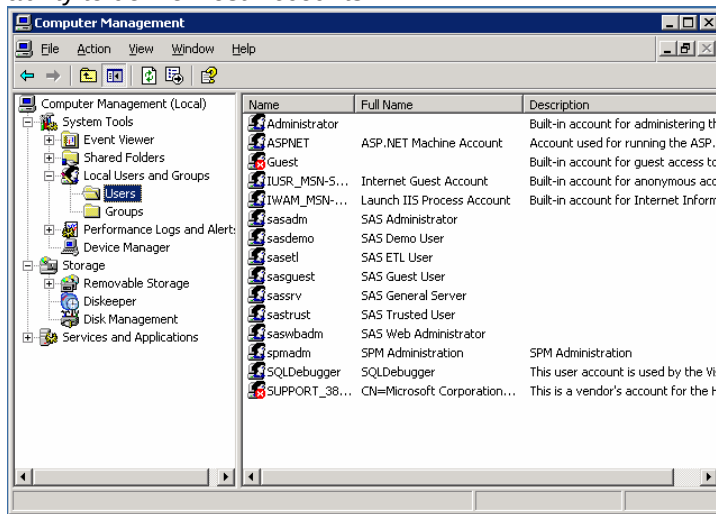
2. How to add a User to the SAS System

This should include instructions related to LDAP/Active Directory group assignments or Host Group setup as well as SAS user administration, such as the User Manager in SAS Management Console or within the SAS web solution such as SAS Warranty Analysis.

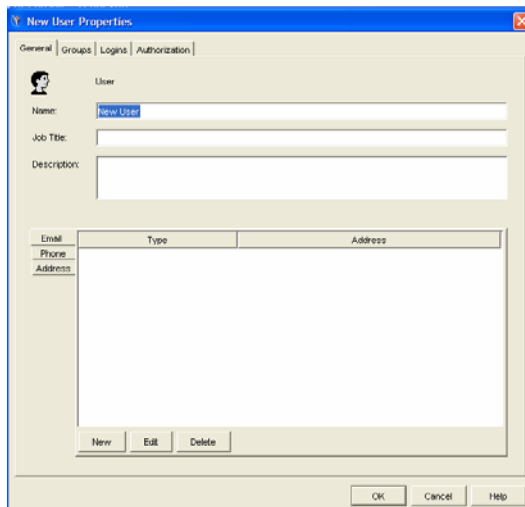
With host authentication and after the user has been defined to the server, the SAS Management Console 'User Manager' plug-in is the only SAS required location to define the

user. Depending on the Security Model in place, the user will need to be assigned to the appropriate Group or Access Control Template (ACT).

In Windows, the Computer Management screen provides the Server Administrator with the ability to define Host Accounts.



In SAS Management Console, the SAS Administrator moves to 'User Manager' and selects 'New User'. After completing the General Information, SAS Administrator will need to update the 'Groups', 'Logins', and 'Authorization' tabs. The key tab is the 'Logins' as this will enable the Host Authentication to work properly. These same steps are available in the Youtube.com video: http://www.youtube.com/watch?v=eFPk_Nlr1Jc



3. Routine/Scheduled Processing

Information should include screenshots, file locations, and other detail to allow for quick problem resolution. Included is a quick list of reference material that would prove helpful to administrators:

- What routine SAS scripts run
- When do they start
- How are they initialized
- When should they be completed
- Where are the logs
- How does one check status

- g. How can a stalled/hung scheduled process be stopped manually

Processing Documentation Example:

Information	ETL Job 1	ETL Job 2
Definition	C:\project\sales\monthlyETL.sas	C:\project\finance\weeklyETL.sas
Schedule	The 3 rd day of each Month	Every Sunday
Executed From	Windows Scheduler	LSF Admin
Completion Timeframe	3:00	0:30
Log Location	C:\project\sales\logs\monthlyETL_&date.log	C:\project\finance\logs\weeklyETL_&date.log
Status Review	Windows Scheduler, Completion Notice Email	Log Location (Size ~ 1MB)
Manually Stop Steps	Stop Job in Windows Scheduler	Task Manager – Kill Process

MANAGE THE SHARED SAS RESOURCES

SAS Temp Space – AKA 'WORK'

In many of the SAS solutions, users have the ability to create their own data, reports, content. SAS also uses swap space on the server to process user requests (such as the WORK library). This information can create space limitation problems. It is important to proactively establish partitioned drives or folders for these files and set up routine scripts to remove older files from the directories. The location of this swap space is found in the SASv9.cfg file. In the below example, this has been changed to the F:\saswork folder.

```
/* Setup the default SAS System user work folder */
/* -WORK "!TEMP\SAS Temporary Files" */
-WORK F:\saswork
```

Clearing Out Old Sessions

Periodic restarts of the SAS Services will ensure effective operation. For instance, SAS Enterprise Guide in a server environment utilizes the SAS Object Spawner. If a user stops a process on their desktop there is the potential that the server's process does not stop. The Object Spawner should be periodically restarted to halt any hung process. It is recommended in a Windows environment that this be done once per week.

Cleaning SAS Temp Space

In the midst of service restarts, clearing out temporary work folders (and contained SAS files) is recommended to reduce space constraints. These temporary work folders utilize the naming convention: "_TD####". This can be set up as an automatic step in the batch process using SAS Functions, PIPE File References and Directory Listings. An example of this is below.

```
/*Windows Utility for Determining Work Directory*/
/*Listing _TD Files Created Before Past Hour*/
/*Creating and Executing Windows Script for Deleting Directories*/

/*Step 1: Determine Location of WORK Directory*/
/* Define FILEREF For Directory Listing*/
data _null_;
val=quote(scan("%sysfunc(pathname(work))", 1, "_"));
call execute("filename direct PIPE 'dir "||strip(val)||"';");
run;

/*Step 2: Define Batch File Location/Name for Cleanup Execution*/
filename batchrm "C:\Support\batchrem.bat";

/*Step 3: Read in Directory List, Clean/Filter for Old _TD */
/* Create Batch File for Cleanup Execution*/
```

```

data _null_;
val=scan("%sysfunc(pathname(work))", 1, "-");
infile direct trunccover;
file batchrm;
input dirlist $255.;
if index(dirlist, '_TD') > 0 then do;
    dir=scan(dirlist, 5, ' ');
    date=input(substr(dirlist,1,10),mmddy10.);
    time=input(substr(dirlist,12,8),time8.);
    format date mmddy10. time hhmm.;
    if date < today() or (date=today() and time < hms(hour(datetime()),
0, 0)) then do;
        put "cd " val;
        put "rmdir " dir "/S /Q";
    end;
end;
run;

/*Step 4: Execute Batch File for Cleanup*/
options noxwait xsync;
x "cd C:\Support";
x "batchrem.bat";

```

Shared Project Space

Utilization is also important for shared resources, as one user could monopolize the server file space and cause other users to experience insufficient file space errors or even a system failure. Disk space can be attributed to users by using a PIPE filename of the shared folder. Sample SAS code for obtaining this information in Windows can be found at:

<http://www2.sas.com/proceedings/forum2008/092-2008.pdf>

Included below is modified code to retrieve the owner information:

```

options noxwait noxsync;
/*step 1: create a directory and size listing of the c:\ drive*/
filename pipedir pipe ' dir "c:\\" /S /Q' lrecl=5000;

/*step 2: read in the directory and parse into directory and date*/
data b;
infile pipedir trunccover;
input line $char1000.;
length directory $1000;
retain directory;
if line = ' ' or
index(uppercase(line), '<DIR>') or
left(uppercase(line))='VOLUME' then
delete;
if left(uppercase(line))='DIRECTORY OF' then
directory=left(substr(line,index(uppercase(line), 'DIRECTORY OF')+12));
if left(uppercase(line))='DIRECTORY OF' then
delete;
if input(substr(line,1,10),?? mmddy10.) = . then
substr(line,1,10)='12/31/2999';
date=input(substr(line,1,10),?? mmddy10.);
format date mmddy10.;
run;

/*step 3: sort by descending date*/
proc sort data=b;
by directory descending date;
run;

/*step 4: create a summary table with file count, size, and ownership*/
data Directory_Summary(drop=line);
set b;
by directory;
length filename $75;
retain owner number_of_files_in_directory directory_size;
if first.directory then

```

```

do;
number_of_files_in_directory=input(scan(line,2,' '),32.);
directory_size=input(scan(line,4,' '),comma32.);
end;
file_size=input(scan(line,4,' '),comma32.);
filename=' ';
owner=input(scan(line,5,' '),$50.);
filename=input(scan(line,6,' '),$50.);
if index(upcase(line),'FILE(S)') then
delete;
if date ge '30DEC2999'd then
delete;
run;

/*step 5: sort in descending file_size*/
proc sort data=directory_summary;
by descending directory_size descending file_size;
run;

/*step 6: further analyze & report to users as needed the results*/
...

```

Log Space Cleanup

It is also advisable to have a log cleanup process in place as log files can quickly max out the available space on the system. Deleting or archiving logs automatically can be accomplished by using a SAS batch script. An example of taking a list of files and using the WINZIP Command Line Utility to compress them into a single ZIP file is available in the user group paper:

<http://www2.sas.com/proceedings/sugi29/077-29.pdf>

LOG REVIEWS

Proactive reviews of all logs should allow for administrators to address issues and perform improvements prior to user discovery. Included are some references for what logs can be utilized and reviewed. Note that while maintaining logs allows for full review of the SAS server's performance and utilization, logs should periodically be archived or removed. Define a policy and create a script to periodically and automatically manage the amount of logs stored.

Scheduled Jobs

Scheduled SAS jobs should include a PROC PRINTTO statement at the beginning to write a permanent log out to a central folder. These should then be reviewed after each run and errors forwarded to the appropriate person via email. There are several SAS User Group Papers to set this up in SAS:

<http://www2.sas.com/proceedings/sugi31/128-31.pdf>

<http://www2.sas.com/proceedings/sugi27/p096-27.pdf>

Server Logs

SAS Server logs are scattered throughout the SAS server configuration. Detail is available in SAS Online Documentation at:

<http://support.sas.com/onlinedoc/913/getDoc/en/bisag.hlp/a003150584.htm>.

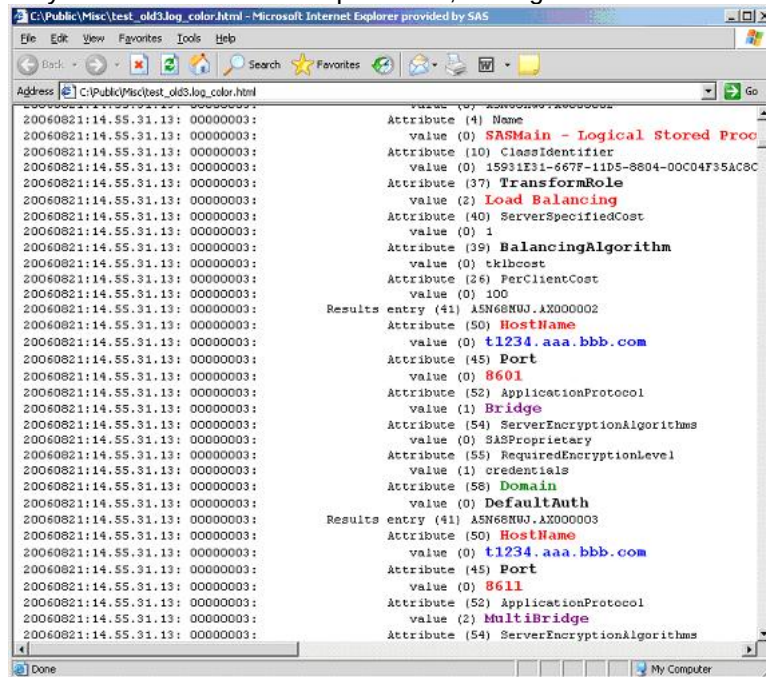
Server	Default Location of Logs Files	Default logging	New log generated every
Metadata Server	SAS-configdir \\Levn\SASMain\MetadataServer\logs\MetadataServer#d#b#y.log	Enabled	Day
Object Spawner	SAS-config-dir \\Levn\SASMain\ObjectSpawner\logs\objspawn.	Disabled	Invocation
Workspace Server	SAS-config-dir \\Levn\SASMain\WorkspaceServer\logs	Disabled	Session

Stored Process Server	SAS-config-dir \Levn\SASMain\StoredProcessServer\logs\Stored_Process_Server_%v.log	Enabled	Session
OLAP Server	SAS-config-dir \Levn\SASMain\OLAPServer\logs\OLAPServer#d#b#y.log	Enabled	Day
SAS/CONNECT Spawner	SAS-config-dir \Levn\SASMain\ConnectServer\logs\spawner.log	Enabled	Invocation

SAS has a Color Coder Utility

<http://support.sas.com/community/newsletters/news/feature/1q2007/bilog.html> to assist with reviewing logs quickly.

As you can see in the example below, the log file is reformatted for review in HTML colors.



User Community Review

Metadata Information can be reviewed to understand the user community, however this only provides the user name and time they connected to a SAS resource. Information on what procedures were run or what databases were accessed will be available in SAS 9.2 with the PROC SCAPROC process.

Within the Color Coder Utility (mentioned above) an Object Spawner Summary tool is available to review Enterprise Guide, Stored Process and Workspace Server usage.

PERFORMANCE MANAGEMENT

Windows environments should be rebooted at least monthly and full defragmentation processes are required to maintain system health. Diskeeper (<http://www.diskeeper.com/defrag.asp>) is a recommended utility that can allow for full boot-time defrags.

Microsoft's Process Monitor (<http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx>) is a GUI interface that allows searching through active DLLs. One can actually search on a scheduled ETL sas job name and find the related sas.exe to monitor, pause or even stop.

A SAS Script can be written and executed during the stop/start services process to ensure that all hung processes are also stopped. Included is an example of this code:

```

/*Review Running SAS Processes*/
/*Filter List for Hung Processes*/
/*Creating and Executing Windows Script for Killing Tasks*/

/*Step 1: List Running Processes Using TASKLIST CMD*/
/*      Filter List of Processes based on Defined Criteria*/
proc printto log='c:\support\logs\hungprocesslisting.log'; run;
filename tasks PIPE 'tasklist /V';
data process;
infile tasks truncover;
input tasklist $255.;
process=scan(tasklist, 1, ' ');
if process='sas.exe' then do;
    PID=scan(tasklist, 2, ' ');
    SessionName=scan(tasklist, 3, ' ');
    if substr(sessionName, 1, 3) = 'RDP' then do;
        SessionName=scan(tasklist, 4, ' ');
        Memory=scan(tasklist, 5, ' ');
        Status=scan(tasklist, 7, ' ');
        UserName=scan(tasklist, 8, ' ');
        CPUtime=input(scan(tasklist,9, ' '),time.);
        WindowTitle=scan(tasklist, 10, ' ')||' '||scan(tasklist,
11, ' ');
    end;
else do;
        SessionID=scan(tasklist, 3, ' ');
        Memory=scan(tasklist, 4, ' ');
        Status=scan(tasklist, 6, ' ');
        UserName=scan(tasklist, 7, ' ');
        CPUtime=input(scan(tasklist,8, ' '),time.);
        WindowTitle=scan(tasklist, 9, ' ')||' '||scan(tasklist, 10,
' ');
    end;
    if Status='Unknown' then do;
        UserName=scan(tasklist, 9, ' ');
        CPUtime=input(scan(tasklist,10, ' '),time.);
        WindowTitle=scan(tasklist, 11, ' ')||' '||scan(tasklist,
11, ' ');
    end;
    format CPUtime time.;
    if UserName not in ('N/A', 'zencos\sasadm') then output;
end;
run;

/*Step 2: Create Batch File for Cleanup Execution*/
filename stplist "c:\support\processstop.bat";
data _null_;
set process;
file stplist;
put 'TASKKILL /PID ' PID ' /T';
run;

/*Step 3: Execute Batch File for Cleanup*/
options noxwait xsync;
x "cd c:\Support";
x "processstop.bat";
x "del /F processstop.bat";
proc printto; run;

```

In UNIX, there are various resources available. The nmon process, explained at SGF2008 provides detail on how to monitor and manage the SAS processes.
<http://www2.sas.com/proceedings/forum2008/387-2008.pdf>

MANAGING INCOMING TICKETS

Of course, the users will still have questions or issues to be addressed. Derive a system for tracking the incoming requests, set expectations for resolving issues and timeliness, and always following up with users as plans are set and when progress is made.

The system could be an email address that multiple people access (so you can take a vacation), or a web database (such as eRoom). Users need to be trained on how to open a ticket, or given a set of guidelines so that the email is addressed quickly. Information should include the user's name and telephone, application used, what was attempted and screenshots.

Another possible and simple mechanism that users could share is a Google Spreadsheet. <http://spreadsheets.google.com/> Is a quick method of tracking and marking off issues as they are being addressed.

The Quality Standard for ISO 9001:2000¹ adds that resolutions should be validated, documented and evaluated for a reduction in reoccurrence. These are common sense steps that allow information to be stored, maintain customer satisfaction, and promote continuous improvements of the system. The important item to remember is to track incoming requests and respond to them!

REFERENCE MATERIAL

Additional Material on Supporting a SAS Server Architecture will be included as new findings are made on Angela's SASCommunity.org page:

~ http://www.sascommunity.org/wiki/Supporting_a_SAS_Server_Architecture

Outside of the URLs included within this paper, the following papers were also referenced:

- ~ Check out These Pipes: Using Microsoft Windows Commands from SAS®, <http://www2.sas.com/proceedings/forum2008/092-2008.pdf>
- ~ How to Avoid Overwriting Work Data Sets and Work Formats, An Essential Macro Design Topic, <http://www2.sas.com/proceedings/forum2008/100-2008.pdf>
- ~ How Do I Look it Up If I Cannot Spell It: An Introduction to SAS® Dictionary Tables, <http://www2.sas.com/proceedings/sugi31/259-31.pdf>
- ~ SAS Paper “**Proven Practices for SAS Intelligence Platform System Administration**”

CONTACT INFORMATION

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

Angela Hall
Zencos Consulting LLC
100 Regency Forest Dr, Suite 130
Cary, NC 27518
Work Phone: 919-459-4600 x116
Angela.Hall@Zencos.com
<http://www.zencos.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.

¹ ISO 9001:2000, Quality Management System Requirements.