Paper A04-2009

No More Blue Screens - Running SAS® on Windows Servers

Joanne Ellwood, Progressive, Mayfield, Ohio

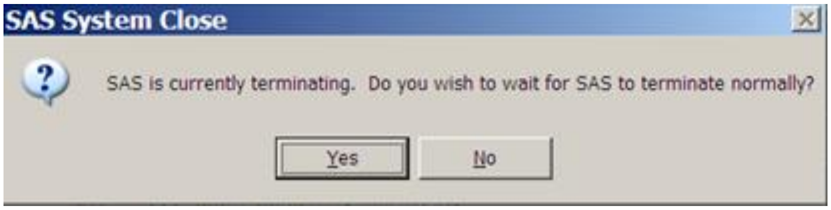
ABSTRACT

Running SAS Foundation products on Windows Servers with many users can be very challenging. Users can experience slowdowns and suspended blue screens when trying to log onto the server. Runaway SAS sessions can consume large amounts of CPU while doing no real work. This presentation will show how SAS server administrators and users can manage these problems using standard Windows utilities, successfully avoiding the dreaded non-scheduled reboot of a SAS application server. Strategies discussed in the presentation may have application to other SAS installations such as BI Server.

MAIN

This presentation assumes running SAS on a dedicated server using the Windows operating system and Terminal Services Management. Multiple users can log onto the server using Remote Desktop Connection and run The SAS System either interactively or via Windows Batch Submit. SAS Foundation products are installed on the server discussed in this presentation; however, strategies used in the presentation may have application to other SAS installations such as BI Server.

In a research environment, SAS users are generally not seasoned developers. User bad behavior such as allowing SAS sessions to remain open on the servers for days or even weeks at a time can strain the server. In a research environment, SAS users may abuse The SAS System by writing and executing code that performs very poor joins and merges. An impatient user may even terminate a SAS session while in remote query without waiting for SAS to terminate normally. We can humbly admit that at one time or another, we have all answered "No" to this question:



SAS is currently terminating. Do you wish to wait for SAS to terminate normally?

However, these types of user bad behavior can adversely impact the performance of the SAS application server.

Windows Servers have best performance when running near or below 50% CPU capacity. When usage gets up to about 80%, users will experience slowdowns in running their SAS programs, in doing simple manipulations of windows and in longer response times when logging onto the server. When CPU usage reaches 100%, the server cannot respond in any manner and users are not able to even log onto the server.

SAS running on dedicated servers will use all CPU resources available. Because of this fact, it is very easy for SAS application servers to quickly reach 80% to even 100% CPU usage.

SAS application server administrators and users can identify runaway SAS sessions because there is a relationship between a SAS job's Process ID (PID) on the server and the _TD foldername in the SAS Temporary Files folder. The PID number and the _TD foldername are the same.

The screenshot below shows Processes detail from the Windows Task Manager sorted by descending CPU Time. On the right side of the screenshot is the folder listing in the SAS Temporary Files folder.

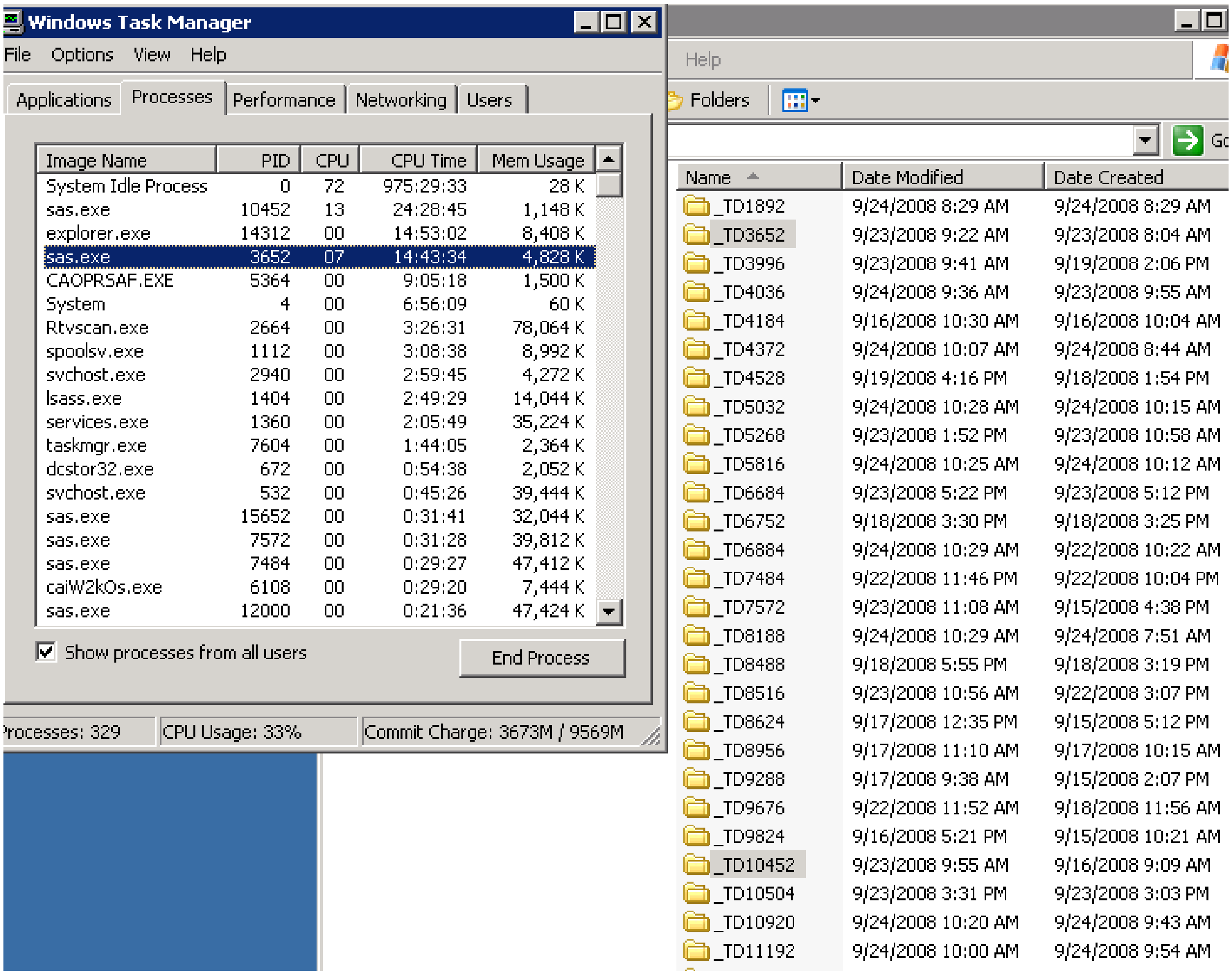


Image Name	PID	CPU	CPU Time	Mem Usage
System Idle Process	0	72	975:29:33	28 K
sas.exe	10452	13	24:28:45	1,148 K
explorer.exe	14312	00	14:53:02	8,408 K
sas.exe	3652	07	14:43:34	4,828 K
CAOPRSAF.EXE	5364	00	9:05:18	1,500 K
System	4	00	6:56:09	60 K
Rtvscan.exe	2664	00	3:26:31	78,064 K
spoolsv.exe	1112	00	3:08:38	8,992 K
svchost.exe	2940	00	2:59:45	4,272 K
lsass.exe	1404	00	2:49:29	14,044 K
services.exe	1360	00	2:05:49	35,224 K
taskmgr.exe	7604	00	1:44:05	2,364 K
dcstor32.exe	672	00	0:54:38	2,052 K
svchost.exe	532	00	0:45:26	39,444 K
sas.exe	15652	00	0:31:41	32,044 K
sas.exe	7572	00	0:31:28	39,812 K
sas.exe	7484	00	0:29:27	47,412 K
caiW2kOs.exe	6108	00	0:29:20	7,444 K
sas.exe	12000	00	0:21:36	47,424 K

☑ Show processes from all users

End Process

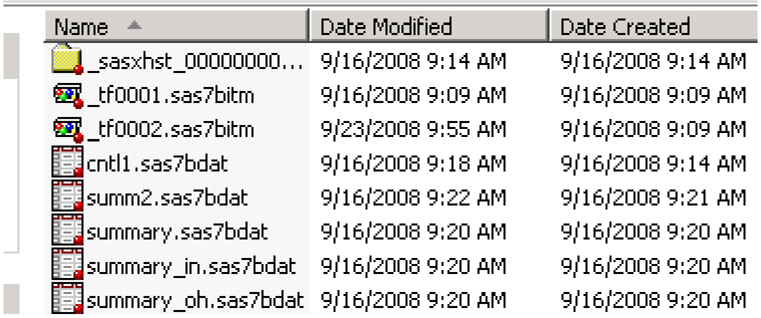
```
rocesses: 329    CPU Usage: 33%    Commit Charge: 3673M / 9569M
```

Name	Date Modified	Date Created
_TD1892	9/24/2008 8:29 AM	9/24/2008 8:29 AM
_TD3652	9/23/2008 9:22 AM	9/23/2008 8:04 AM
_TD3996	9/23/2008 9:41 AM	9/19/2008 2:06 PM
_TD4036	9/24/2008 9:36 AM	9/23/2008 9:55 AM
_TD4184	9/16/2008 10:30 AM	9/16/2008 10:04 AM
_TD4372	9/24/2008 10:07 AM	9/24/2008 8:44 AM
_TD4528	9/19/2008 4:16 PM	9/18/2008 1:54 PM
_TD5032	9/24/2008 10:28 AM	9/24/2008 10:15 AM
_TD5268	9/23/2008 1:52 PM	9/23/2008 10:58 AM
_TD5816	9/24/2008 10:25 AM	9/24/2008 10:12 AM
_TD6684	9/23/2008 5:22 PM	9/23/2008 5:12 PM
_TD6752	9/18/2008 3:30 PM	9/18/2008 3:25 PM
_TD6884	9/24/2008 10:29 AM	9/22/2008 10:22 AM
_TD7484	9/22/2008 11:46 PM	9/22/2008 10:04 PM
_TD7572	9/23/2008 11:08 AM	9/15/2008 4:38 PM
_TD8188	9/24/2008 10:29 AM	9/24/2008 7:51 AM
_TD8488	9/18/2008 5:55 PM	9/18/2008 3:19 PM
_TD8516	9/23/2008 10:56 AM	9/22/2008 3:07 PM
_TD8624	9/17/2008 12:35 PM	9/15/2008 5:12 PM
_TD8956	9/17/2008 11:10 AM	9/17/2008 10:15 AM
_TD9288	9/17/2008 9:38 AM	9/15/2008 2:07 PM
_TD9676	9/22/2008 11:52 AM	9/18/2008 11:56 AM
_TD9824	9/16/2008 5:21 PM	9/15/2008 10:21 AM
_TD10452	9/23/2008 9:55 AM	9/16/2008 9:09 AM
_TD10504	9/23/2008 3:31 PM	9/23/2008 3:03 PM
_TD10920	9/24/2008 10:20 AM	9/24/2008 9:43 AM
_TD11192	9/24/2008 10:00 AM	9/24/2008 9:54 AM

Two SAS sessions in the Windows Task Manager appear to be runaway SAS sessions because they both have large CPU Time usage. The SAS Temporary Files folder shows that the PID 3653 SAS session associated with the _TD3652 SAS temporary folder was recently created and has had recently been active (determined from Date Created and Date Modified in the folder listing). Also the session has relatively low CPU usage of 7%. This session looks to be a valid active SAS session.

However, the SAS Temporary Files folder shows that the PID 10452 SAS session associated with the _TD10452 SAS temporary folder was created roughly a week ago and looks to have had some recent activity per its Date Modified. In looking at the contents of the _TD10452 folder in the screenshot below, one can see that there really hasn't been any recent user activity in the SAS session. For these reasons, this session is suspected to be a runaway SAS session.

_TD10452



Name	Date Modified	Date Created
_sasxhst_00000000...	9/16/2008 9:14 AM	9/16/2008 9:14 AM
_tf0001.sas7bitm	9/16/2008 9:09 AM	9/16/2008 9:09 AM
_tf0002.sas7bitm	9/23/2008 9:55 AM	9/16/2008 9:09 AM
cntl1.sas7bdat	9/16/2008 9:18 AM	9/16/2008 9:14 AM
summ2.sas7bdat	9/16/2008 9:22 AM	9/16/2008 9:21 AM
summary.sas7bdat	9/16/2008 9:20 AM	9/16/2008 9:20 AM
summary_in.sas7bdat	9/16/2008 9:20 AM	9/16/2008 9:20 AM
summary_oh.sas7bdat	9/16/2008 9:20 AM	9/16/2008 9:20 AM

One way to confirm that the SAS session is a runaway is to observe the PID in the Windows Task Manager. A runaway SAS session will not have much change in % CPU usage. In order to stress a server, only a handful of runaway SAS sessions running between 10-15% can do the trick. Another way to confirm that a SAS session is a runaway is to ask the user of the session. The user is easily identified as the User Name in the Windows Task Manger or as the folder Owner in the SAS Temporary Files folder listing. For confidentiality purposes, both columns have been omitted from this presentation's screenshots. Once a SAS session has successfully been identified as being a runaway process, the administrator or the user can end the SAS process in the Task Manager.

If the server is running at 100%, even the administrator may not be able to log onto the SAS application server. In such cases, pstools (a Windows freeware) can be used to execute the same process. The steps to do so are:

1. Map to the server's SAS Temporary Files folder.
2. In a command window, use the pslist command to see all SAS processes that are currently running – **plist \\*servername* –e sas**.
3. Compare the PID's of high CPU Time usage sessions to the _TD files folders and determine which processes can be killed.
4. Since server is running at 100% and no work can be done, killing suspected runaway SAS Sessions is a more acceptable alternative to rebooting the server which would kill everyone's SAS sessions. The pstools command to kill a specific SAS session is **pskill \\*servername* –t *pid***.

In our research environment, we recommend the following practices for our SAS servers users:

1. Log off the server when finished running SAS programs and don't stay logged onto the server for days at a time.
2. Use Microsoft "Batch Submit" to run a SAS program that has stabilized through the analytic process. Microsoft Batch Submit creates in the current folder the log and listing files with the same names as the SAS program file.
3. Save critical SAS datasets to disk and close all interactive SAS sessions before logging off the server each day.
4. Clean up temporary (work) SAS datasets in the SAS session that are no longer needed. This can be done interactively in the SAS Explorer window or in the SAS program with drop/delete table.

CONCLUSION

A SAS application server administrator cannot always prevent user bad behavior from adversely affecting the performance of the server. In situations where the server is stressed to the point of being unresponsive, the administrator and the server users may be able to avoid a non-scheduled reboot of the server by leveraging information gathered from the Task Manager, the SAS Temporary Files folder and pstools to kill runaway SAS processes.

CONTACT INFORMATION

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

Joanne Ellwood
Progressive
Mayfield, Ohio
440.395.9372