Paper 215-29

Measuring SAS® Software Usage On Shared Servers With the RTRACE Facility

Michael A. Raithel, Westat, Rockville, MD

Abstract

How often is SAS software being used in your organization, who is using it, and which SAS products are being used? These questions are not always easy to answer, particularly for organizations that run SAS on multiple platforms. However, an innovative technique for exploiting SAS RTRACE files can provide you with a way to record and report information about SAS usage in your organization. This paper presents a methodology for exploiting SAS's native RTRACE facility to track all SAS batch and interactive sessions in the Windows, Unix, and Linux environments. It provides an overview of the RTRACE facility and how to set it up to run behind-the-scenes in shared server environments. The paper contains example programs that process RTRACE files and store the information in SAS data sets. It includes a sample program that reports SAS usage based on the captured RTRACE information. After reading this paper, you will have all of the tools that you need to exploit RTRACE, and determine the "who", "what", "when" and "where" of SAS software usage in your own organization.

Introduction

Not surprisingly, most SAS programmers have never heard of the RTRACE facility. However, it has been around for a while and is available for SAS running in the Windows®, Unix, Linux, OS/2, and CMS environments. The RTRACE facility is well documented in the SAS Online Documentation. If you access the SAS Online Documentation, simply perform a search on "rtrace" to find the relevant sections. The online document, *Creating a Scaled Down Version of the SAS System for Distribution*, provides an especially helpful overview of the RTRACE facility.

The RTRACE facility was originally created so that SAS users may:

- Create and distribute a subset of SAS to licensed users at their site who run a particular SAS application.
- Create scaled-down versions of SAS for use on computers for which disk space is scarce.
- Customize the global installation of SAS at their site to optimize use of disk space on individual machines.

SAS users can perform the above functions by first invoking the RTRACE facility and then running a SAS application. The RTRACE facility will create an RTRACE file that contains a record of every SAS system file that was opened during the execution of the SAS application. Then, SAS users can employ the %COPYSAS automatic SAS macro to read the RTRACE file and create a scaled down version of SAS that will be able to run that particular application. The scaled down version may then be ported to other computers so that the application can be run on them.

You can create an RTRACE file by using the *rtrace* and the *rtraceloc* system options when invoking SAS. The best place to specify these options is in the SAS configuration file. Here is an example of how they may be coded:

```
-rtrace all  -rtraceloc f:\rtrace\rtracefile.txt
```

In the example, above, *-rtrace all* specifies that SAS is to list all of the files used in the SAS session. The *–rtraceloc* option tells SAS the name of the file to which RTRACE information is to be written. In this example, RTRACE information will be written to the "*f:\rtrace\rtracefile.txt*" file. If that file did not exist prior to the invocation of the SAS session, it will be created; if it did exist, it will be overwritten with information from the latest SAS session. Note that both options begin with a hyphen ("-") because they are being specified in the SAS configuration file.

After the SAS session has completed, the data in the RTRACE file can be browsed using common system tools such as WordPad on Windows and VI on Unix. Here is an example of the first few lines of an RTRACE file created by SAS V8.2 running on the Windows operating system:

```
File closed: C:\progra~1\sasins~1\sas\v8\autoexec.sas
File opened: F:\sas82\SAS\core\sasexe\saszaf.DLL
File referenced: F:\sas82\SAS\core\sasexe\sasfm.DLL
File referenced: F:\sas82\SAS\core\sasexe\sasfm.DLL
File opened: F:\sas82\SAS\core\sasexe\sasfm.DLL
File opened: F:\sas82\SAS\core\sasexe\sasp14.DLL
File referenced: F:\sas82\SAS\core\sasmsg\base.msg
File opened: F:\sas82\SAS\core\sasmsg\base.msg
```

As you can see in the example above, the files that the SAS System opens and closes are recorded in the RTRACE file. That is the key to using RTRACE entries to identify SAS product usage. Files for a particular SAS product will only be opened when that product is invoked during the course of the SAS session. By parsing an RTRACE file and looking at which SAS product files have been opened, you can discern which SAS products were used during the course of the SAS program's execution. If you are a SAS Administrator running SAS on a shared server, such as a Windows server or a Unix server, this information can give you an invaluable look at exactly which SAS products your users are really accessing.

The author developed RTRACE-based SAS usage reporting systems on Windows, Unix, and Linux servers. SUGI paper size limitations make it impractical to thoroughly discuss all three systems. So, this paper focuses on the system developed for Windows servers. Keep in mind that many of the concepts and strategies outlined in this paper also apply to Unix and Linux servers. The author will gladly reply to requests for information about setting up and running a system on Unix and Linux servers (Refer to the author's Contact Information at the end of this paper).

The following sections of this paper describe the specific implementation of an RTRACE-based SAS usage reporting system on Windows servers. **Appendix A**, at the end of the paper, contains the SAS programs used to collect, store, and report on SAS usage information. **Appendix B** contains a portion of a CONTENTS procedure listing of the Windows production RTRACE SAS data set. And, **Appendix C** contains brief sections of some of the RTRACE reports.

A Methodology for Collecting and Processing RTRACE Data

Once you have determined how to create RTRACE files, you will want to institute a methodology for collecting and processing them. Creating and collecting RTRACE data should be done in an unobtrusive manner that is transparent to users. By automatically invoking the RTRACE facility behind-the-scenes, without user intervention, you can be sure of consistent data collection from all SAS sessions running on your shared servers. This section describes five basic elements of a successful RTRACE-based SAS usage reporting system.

1. Transparent recording of RTRACE information for every user's interactive and batch SAS sessions

Transparent recording of RTRACE information is necessary because you do not want to put the burden of initiating the RTRACE facility on individual users. Collecting RTRACE-based SAS usage information is not their responsibility and would cause them to have to carry out an extra step when performing their programming tasks. There would be a number of users who would forget to initiate the RTRACE facility, so you would not get an entirely accurate picture of what is happening with SAS software in your organization. That is why it is imperative that the recording of RTRACE information be performed transparently, behind-the-scenes, for all users.

Invoking the RTRACE facility does not have an appreciable performance impact on the SAS session in program elapsed time, CPU time, or disk usage. So it is unlikely that most users would even notice that it has been employed.

2. One separate RTRACE file created for every interactive and batch SAS session executed on the server

One separate RTRACE file should be created for every interactive and batch SAS session that is executed.. This is necessary because an individual user could have several batch SAS sessions running while simultaneously working in an interactive SAS Display Manager session. An attempt to have all of these SAS sessions writing to the same RTRACE file would overwrite information in the file. Also, the date/time embedded into the file name is crucial to determining when, exactly, a SAS session began. So, it is necessary to have one RTRACE file per SAS session; even if that results in hundreds of RTRACE files being created in a single day. The RTRACE naming conventions for Windows servers, described below, result in RTRACE files with unique names.

3. The ability to identify each user's RTRACE files and the date/time that they were created
Another important element of an RTRACE-based SAS usage reporting system is the ability to uniquely identify each user's RTRACE information. By doing so, you can determine how many individual users you have working with SAS products at a given time. This information can be useful in determining if a particular SAS product (e.g. SAS/IML or SAS/STAT) is receiving heavier use than expected. That is why Userid was made a part of the RTRACE file name in our implementation of an RTRACE-based SAS usage information system.

4. Daily retrieval, processing and deletion of all RTRACE files created the previous day
Daily retrieval, processing, and deletion of RTRACE files created on the previous day is the key to harvesting RTRACE information. You will want to set up automatic programs that read both the contents (to determine SAS Products used) and the names of the RTRACE files to glean Product, Userid, Start Date, Start Time, End Date, and End Time information. Once you have that information, the RTRACE file is no longer of any use and can be deleted. If you have a large number of SAS users, you will find that your RTRACE file directory will contain scores or hundreds of RTRACE files at the end of a day. Therefore, processing them on a daily basis will be necessary to ensure good disk space management.

5. Storage of RTRACE data in a permanent RTRACE SAS data set
When collected, the RTRACE information should be stored in a permanent SAS data set. That data set will act as a database and become the central repository of RTRACE-based SAS usage information. Doing this will allow you to collect data over time and have it on-hand in order to report SAS usage trends. Then, you can craft report programs that read the permanent RTRACE SAS data set and produce reports that are useful to your organization.

Exploiting RTRACE Files

The standard RTRACE file does not contain very much useful information. It simply holds a list of files opened and closed by the SAS system. SAS does not provide an exit whereby users may record additional information in an RTRACE file. Consequently, to use the RTRACE facility as the basis for a SAS usage reporting system, it will be necessary to perform some behind-the-scenes programming in order to capture all of the information that you need.

Here is the type of information that is useful to have to measure SAS usage:

- **Userid** – The User Id of the person executing SAS
- **Start Date** – The date on which the SAS session was started
- **Start Time** – The time that the SAS session was started
- **End Date** – the date on which the SAS session ended
- **End Time** – the time that the SAS session ended
- **Product** – the SAS product that was used
- **System ID** – The system on which SAS was executed—if your organization has multiple SAS servers

Several methods were used to obtain the information listed above on the various platforms. The Userid, Start Date, and End Date were made a part of the RTRACE file name. The End Date and End Time were obtained from the operating system's recording of the date/time that the RTRACE file was last modified. (This translates to the date/time that the RTRACE file was closed at the end of the SAS session). Product was gleaned from processing the data within the RTRACE file. Finally, the System ID was passed to the RTRACE data collection program via a SAS macro variable. This was necessary because System ID is an organizational designation, and not available programmatically on the various platform's operating systems.

The naming convention for RTRACE files on Windows servers was the concatenation of:

- Userid
- Start Date
- Start Time
- The ".rtd" file extension

On a Windows server, a typical RTRACE file looks like this:

```
raithel_m27APR2003073015.rtd
```

In the example, above, the Userid is “raithel_m”, the Start Date is April 27, 2003, and the Start Time is 7:30:15 a.m. If Windows Explorer lists the “Date Modified” as “04/27/2003 7:42 AM”, then the End Date is April 27, 2003 and the End Time is 7:42 a.m. Because this RTRACE file is found in the RTRACE file directory on the SYSTEM1 server, the System ID is “SYSTEM1”.

Creating RTRACE Files on Windows Servers

Creating RTRACE files behind-the-scenes for all users of SAS on a Windows server is a challenge. It is a challenge because the *rtrace* option must be specified in the user’s SAS configuration file. A user’s SAS configuration file is static, so whatever values you code in it will remain constant for every SAS interactive and batch session the user initiates. Therefore, if you were to code the following in a particular user’s SAS configuration file

```
-rtrace all  -rtraceloc f:\rtrace\raithel_m_rtracefile.txt
```

then all of that user’s SAS sessions would attempt to write to the same file. The file would be constantly overlaid with RTRACE information from the last SAS session. Consequently, you would not obtain an individual RTRACE file for each SAS session the user initiated.

This problem was resolved by noting the fact that the *rtraceloc* option may be specified in an OPTIONS statement. Therefore, *rtraceloc* can be specified in the *autoexec.sas* file, which is executed directly after SAS initialization—after options in the *sasv8.cfg* file have been set by SAS.

Here are the first few lines of the *sasv8.cfg* file that we supply to all users:

```
/*RTRACESTART-Do not remove the following settings***/

-rtrace all
-rtraceloc "C:\progra~1\sasins~1\sas\v8\rtrace.rtd"
-autoexec "C:\progra~1\sasins~1\sas\v8\autoexec.sas"

/*RTRACESTOP*******************************************/
```

In the example, above, the RTRACE facility is enabled by the *-rtrace all* option. The *-rtraceloc* option is hard-coded to point to a file on the user’s “C” drive. Initial RTRACE messages will be written to this default file until the *autoexec.sas* file is executed. As you will see, the *autoexec.sas* file has SAS code in it that creates a new, unique RTRACE file and executes the *rtraceloc* option again to point to it. Afterward, all subsequent RTRACE messages are written to the new, unique RTRACE file. Since SAS code in the *autoexec.sas* file executes prior to SAS programs run by a user, you can be sure that all SAS products used during the execution of the user’s programs will be recorded in the new RTRACE file.

If *-rtraceloc was not* specified in the *sasv8.cfg* file, then the initial RTRACE messages would be written to the user’s SAS log. This would be confusing to the user and would not make the RTRACE facility very transparent.

Here are the lines in the *autoexec.sas* file that create a unique RTRACE file and enable it for use by the current SAS session:

```
/*rtracebegin**********Do Not Remove**********************/

%let userid = %sysget($USERNAME);
%let _utime =%sysfunc( datetime(), datetime19.0 );
%let usertime =%sysfunc(compress(&_utime,':'));
options  rtraceloc="F:\sastrace\&userid.&usertime..rtd";

/*rtraceend********************************************/
```

In the code, above, the Userid and the current date/time are retrieved from the system. The date/time variable is compressed so that there are no blanks within it. Then, an OPTIONS statement is executed with the *rtraceloc* option so that the new RTRACE file is created in the corporate RTRACE directory: "*F:\sastrace*". The RTRACE file name is composed of the Userid concatenated to the date/time, concatenated to the ".rtd" suffix. After the OPTIONS statement executes in the *autoexec.sas* file, all RTRACE statements generated by statements within the newly initiated SAS session will be written to the new RTRACE file. Here is what the full path of the typical RTRACE file on one of our SAS Windows servers looks like:

```
        F:\sastrace\raithel_m27APR2003073015.rtd
```

Identifying SAS Products on Windows Servers

Once an RTRACE file has been created, you must parse it to determine which SAS products were used. But, with hundreds of lines specifying files that were opened and closed in an RTRACE file, that is not necessarily a straight-forward task. If you were to browse the folders for an individual SAS product on your Windows server, for instance ***!sasroot*\core\sasexe** or ***!sasroot*\stat\sasexe**, you would find scores of ".dll" files. It would take some effort to gather all of them together and put them into a usable form—such as a SAS format—to use within your program to check for the use of BASE SAS or SAS/STAT, or any other products that you might have.

The author noted that when a particular SAS product is used in a SAS session, SAS always opens *.dll* files in the "*sasexe*" folder for that product. For instance, when SAS/STAT is used, *.dll* files in directory ***!sasroot*\stat\\<u>sasexe</u>** are opened so that they may be executed by SAS. So, it is a simple process to parse the RTRACE file looking for lines containing the value of "**\sasexe\**". When that value is found, a look at the previous directory node will yield the SAS product that was used in the SAS session. For example, consider the following line from an RTRACE file:

```
        File opened: F:\sas82\SAS\graph\sasexe\sasamgis.dll
```

The SAS program that parses the RTRACE file would find "\sasexe\" on that particular line and determine that this line is of interest. It would then look at the previous directory node, which is "**graph**". The SAS program would take the value "**graph**" and store it in the Product variable of the RTRACE SAS data set that is being created. (That program, *Process_Rtrace.sas,* may be found in **Appendix A**).

Any individual SAS product (such as SAS/GRAPH) may be used more than once during the course of a SAS session. However, there is no way to associate a particular time stamp with exactly when the product was used. One can only conclude that it was used multiple times within the same SAS session between the Start Date/Start Time and the End Date/End Time. Complicating matters further is the fact that SAS may have used multiple *.dll* files in a single execution of a particular SAS product. So, it does not make sense to generate more than one observation for each product used during a SAS session. Consequently, the SAS program that parses Windows RTRACE files performs a PROC SORT with the NODUPREC option to remove duplicate observations of a particular SAS product found in a given RTRACE file. This means that the final SAS data set created by parsing an RTRACE file has one observation per unique SAS product used during the execution of the SAS session.

Using the "File opened" message, above, and the RTRACE file name (*raithel_m27APR2003073015.rtd*) from the previous section, we would get one SAS observation created for SAS/GRAPH. That observation would have the following values:

- **Userid** – Raithel_m
- **Start Date** – April 27, 2003
- **Start Time** – 07:30.15
- **End Date** – April 27, 2003
- **End Time** – 07:42
- **Product** – graph
- **System ID** – SYSTEM1

Processing RTRACE Files on Windows Servers

After you have modified the *autoexec.sas* and *sasv8.cfg* files so that RTRACE files are created for SAS sessions, you will be ready to process those RTRACE files. This can be done by scheduling a *.bat* file to execute the SAS program that processes RTRACE files on a daily basis from a Windows server. As mentioned previously, the SAS program should capture RTRACE information from each RTRACE file, write that information to the permanent RTRACE SAS data set, and delete the RTRACE file. This section describes the programs we implemented to carry out those tasks. A copy of each program may be found in **Appendix A**.

Daily_Data_Capture.bat
This *.bat* file is scheduled to execute just after midnight, each morning, by a Windows server's Task Scheduler. The Windows server that it runs on is a central server that can access all of the other Windows servers that SAS is run on. So, the RTRACE files in each SAS server's RTRACE directory (e.g. *\\system1\sys\rtrace*) can be reached. This *.bat* file executes the RTRACE SAS driver program, *Rtrace_Driver_Program.sas*, a sparse SAS program that %INCLUDEs and executes the various RTRACE file processing programs.

Rtrace_Driver_Program.sas
This is a bare-bones SAS "shell" program that %INCLUDEs the RTRACE file processing programs—*Identify_Rtrace_Log_Files.sas* and *Process_Rtrace.sas*—and executes them. Both of the afore-mentioned programs are SAS Macro programs. The driver program invokes the Macro programs for each Windows server that has an RTRACE directory on it. It specifies Macro variables that identify the relative directories (e.g. *\\system1\sys\rtrace*) where the RTRACE files can be found on each server when it invokes the SAS Macro programs. The simple design of the driver program allows one to easily add and remove SAS Windows servers that are to have their RTRACE files processed.

Identify_Rtrace_Log_Files.sas
This SAS program contains the %IDENTIFY SAS Macro that reads an RTRACE directory (e.g. *\\system1\sys\rtrace*) and creates invocations of the %RTRACERD SAS Macro that is contained in the *Process_Rtrace.sas* program. It does this for each RTRACE file that it finds in the specified directory. If there were, for example, 200 RTRACE files in the *\\system1\sys\rtrace* RTRACE directory, this program would generate 200 invocations of the %RTRACERD SAS Macro—one for each file So, the end product of executing *Identify_Rtrace_Log_Files.sas* is a file full of SAS Macro program invocations that look like this:

```
        %RTRACERD(Q:\data,\\system1\sys\sastrace\raithel_m27APR2003073015.rtd);
```

The *Identify_Rtrace_Log_Files.sas* program's %IDENTIFY Macro accepts two parameters: 1. The location of the RTRACE permanent SAS data set (e.g. *Q:\data*) , and 2. The full path, relative directory name of the RTRACE directory for a given SAS server (e.g. *\\system1\sys\sastrace*).

Process_Rtrace.sas
This is the program that actually processes the contents of the RTRACE files. It contains the %RTRACERD SAS Macro that accepts two parameters: 1. The location of the RTRACE permanent SAS data set (e.g. *Q:\data*) , and 2. The full path, relative directory name of an RTRACE file (e.g. *\\system1\sys\sastrace\raithel_m27apr2003073015.rtd*). This program performs the following functions:

- This program first determines if the RTRACE file is still being used by a program. If it is, then the RTRACE file is not processed—because it is still being updated—and a message is written to the SAS log. If it is not in use, the RTRACE file is processed.
- Next, the program invokes the *PC_File_Date_Time_Routine.sas* program. This is a SAS Macro program that accepts the full path file name of an RTRACE file and returns two Global SAS Macro variables: 1. FILEDATE—which is the date the RTRACE file was last updated, and 2. FILETIME—which is the time the RTRACE file was last updated. The value of FILEDATE becomes the RTRACE observation's End Date, and the value of FILETIME becomes the RTRACE observation's End Time when returned to the Process_Rtrace.sas program. A copy of the *PC_File_Date_Time_Routine.sas* program may be found in Appendix A.
- This program continues by obtaining the System ID from the first node of the relative directory name passed to it. For example, the System ID for RTRACE file *\\system1\sys\sastrace\raithel_m27apr2003073015.rtd* would be "**system1**".
- Next, the program reads every line in the RTRACE file, deleting lines that do not contain ".DLL" and making observations out of the ones that do. The resulting SAS file is sorted by Start Date, Start Time, System ID, User ID, and Product, using the NODUPKEY SAS sort option. This ensures that there is one observation per Product for this particular SAS session.
- Finally, this program writes the newly created observations to the RTRACE permanent SAS data set.

Creating SAS Resource Usage Reports

The reward for implementing an RTRACE-based SAS usage reporting system is being able to fully characterize and analyze the use of SAS software in your organization. Once you have a good amount of RTRACE information collected in a permanent SAS data set, it is easy to create reports that highlight the information. You may choose to have reports scheduled to run on a cyclic basis—perhaps once a month—or run SAS usage reports on an ad-hoc basis. If you have SAS/IntrNet software, it is simple to create a web-based SAS usage reporting system.

There are a number of ways that you can characterize SAS usage: Concurrent SAS Users by Hour, Total SAS Users for Each SAS Product, Total Unique SAS Users by Date, are just a few. How you report on the information you collect will depend upon what is important to your organization. The *Rtrace_Reports.sas* program found in **Appendix A** provides a good starting point for your own SAS usage reports. **Appendix C** contains brief sections of some of the RTRACE reports created by *Rtrace_Reports.sas*.

Conclusion

Though the RTRACE facility was originally designed to provide a scaled-down version of SAS, it can be used to create a SAS usage reporting system for SAS servers. You will want to implement such a system in a non-obtrusive, systematic manner so that you get SAS usage information from every SAS interactive and batch session initiated on your organization's SAS servers. It will be necessary to "add value" to the RTRACE files by recording information other than the specific files SAS opened and closed such as User ID, Start Date, etc. Once the files are created, they can be harvested on a daily basis by automated SAS programs. The final result of all of this effort is a SAS usage information database from which you may create reports that help you to fully characterize the use of SAS software in your organization.

DISCLAIMER: The contents of this paper are the work of the author and do not necessarily represent the opinions, recommendations, or practices of Westat.

References

SAS OnlineDoc®, Version 8, Copyright 2000, SAS Institute, Inc.

Acknowledgements

I would like to acknowledge Greg Cox, from Westat, who came up with and implemented the clever method of coding the *rtrace* and *rtraceloc* options in both the *sasv8.cfg* and the *autoexec.sas* files.

Contact Information

Please feel free to contact me if you have any questions or comments about this paper. You may reach me at:

Michael A. Raithel
Westat
1650 Research Boulevard
Room RW4521
Rockville, Maryland 20850
michaelraithel@westat.com

Appendix A. - Windows Servers RTRACE Programs

This appendix contains the SAS programs used for processing and reporting on RTRACE files created on Windows servers.

Daily_Data_Capture.bat

```
"C:\Program Files\SAS Institute\SAS\V8\SAS.EXE"  -icon -noterminal -nosplash
-noxwait -noxsync
-CONFIG "C:\Program Files\SAS Institute\SAS\V8\Sasv8.cfg"
-SYSIN "Q:\programs\Rtrace_Driver_Program.sas"
-LOG  "Q:\programs\Rtrace_Driver_Program.log"
```

Rtrace_Driver_Program.sas

```
*****************************************************************************;
*                                                                           *;
* Program: Rtrace_Driver_Program.sas                                        *;
*                                                                           *;
* Purpose: This program %INCLUDEs several RTRACE SAS Macro programs and     *;
*          executes them.                                                   *;
*****************************************************************************;

*****************************************************************;
* SAS Options, Formats, Includes, etc.                          *;
*****************************************************************;
options noxwait xmin xsync;

*************************************************************************;
* INCLUDE the SAS Macro that reads date/time the RTRACE file was closed.*;
*************************************************************************;
%include 'Q:\programs\PC_file_date_time_routine.sas';

*************************************************************************;
* INCLUDE the SAS Macro program that processes RTRACE files.            *;
*************************************************************************;
%include 'Q:\programs\Process_Rtrace.sas';

*************************************************************************;
* INCLUDE the SAS Macro program that identifies RTRACE files and builds *;
* builds macro calls that execute Process_Rtrace.sas for each file.     *;
*************************************************************************;
%include 'Q:\programs\Identify_Rtrace_Flat_Files.sas';

*************************************************************************;
* Execute the SAS programs that Identify and Process the RTRACE files.  *;
*************************************************************************;

%IDENTIFY(Q:\data,\\system1\sys\sastrace);         /*  \\system1\sys  */
%IDENTIFY(Q:\data,\\system2\sys\sastrace);         /*  \\system2\sys  */
%IDENTIFY(Q:\data,\\system3\sys\sastrace);         /*  \\system3\sys  */
```

Identify_Rtrace_Log_Files.sas

```
*****************************************************************************;
* Program: Identify_Rtrace_Log_Files.sas                                    *;
*                                                                           *;
* Purpose: This program reads all of the file names in a directory and      *;
*          creates calls to the RTRACERD SAS Macro, located in the          *;
*          Process_Rtrace_Flat_Files.sas program.                           *;
*                                                                           *;
*          This Macro program accepts two parameters:                       *;
*                                                                           *;
*            SASLIB - This is the full path name of the RTRACE production   *;
*                     SAS data set. This value will be coded into the call  *;
*                     for the RTRACERD SAS macro, that is the output of     *;
*                      this program.                                       *;
*             DIRNAME - The full path name of the directory that contains the*;
*                       RTRACE flat files.                                 *;
*                                                                          *;
*             NOTE: This program expects that all RTRACE flat files will   *;
*                   have an extension of '.rtd'!                           *;
****************************************************************************;

  /*****************************************************/
  /* Beginning of the IDENTIFY SAS Macro.             */
  /*****************************************************/
%MACRO IDENTIFY(SASLIB,DIRNAME);

*******************************************************************;
* Pipe the names of RTRACE files to the SAS System for processing. *;
*******************************************************************;
filename dircmd pipe "dir /b &DIRNAME";

*******************************************************************;
* Temporary file to hold the RTRACERD SAS Macro call statements.   *;
*******************************************************************;
filename holdmacs TEMP;

*******************************************************************;
* Process each specific file name, dropping unneeded ones. Format  *;
* SAS RTRACERD Macro calls for valid RTRACE files.                 *;
*******************************************************************;
data _null_;

length outline $100;

file holdmacs;

infile dircmd missover length=length;

input @;

input bigline $varying200. length;

x = index(bigline,'.rtd');

if x = 0 then delete;

bigline = trim(bigline);

outline = '%RTRACERD(' || "&SASLIB" || ',' || "&DIRNAME" || '\' || trim(left(bigline))
|| ');' ;

put outline;

run;

*******************************************************************;
* Include holdmacs, which is now a series of RTRACERD SAS Macro    *;
* invocations. This will execute Process_Rtrace.sas.               *;
*******************************************************************;
%INCLUDE holdmacs;
```

```
  /****************************************************/
  /* End of the IDENTIFY SAS Macro.                   */
  /****************************************************/
%MEND IDENTIFY;
```

<u>PC_File_Date_Time_Routine.sas</u>

```
*******************************************************************************;
* Program: PC_File_Date_Time_Routine.sas                                      *;
*                                                                             *;
* Purpose: This program gets the system date and time for a specified PC file.*;
*          The program is in the form of a SAS Macro, named PCFLDTTM, and has *;
*          one parameter that must be specified:                              *;
*                                                                             *;
*             PCFILE - The full directory path to the PC file. For example:   *;
*                                                                             *;
*                       %PCFLDTTM(c:\windows\temp\myfile.txt);                *;
*                                                                             *;
*           The output of this program is two global SAS Macro variables:     *;
*                                                                             *;
*           FILEDATE - Contains the date of the PC file in the SAS date9.     *;
*                      format. Here is an example of how FILEDATE might be    *;
*                      specified in a DATA step:                              *;
*                                                                             *;
*                          filedate = "&FILEDATE"d;                           *;
*                                                                             *;
*           FILETIME - Contains the time of the PC file in the SAS time5.     *;
*                      format. Here is an example of how FILETIME might be    *;
*                      specified in a DATA step:                              *;
*                                                                             *;
*                          filetime = "&FILETIME"t;                           *;
*******************************************************************************;

%MACRO PCFLDTTM(PCFILE);

%GLOBAL FILEDATE FILETIME;

*********************************************************************;
* Pipe a system dir /4 command to filename EXTFILE.                 *;
*********************************************************************;
filename extfile pipe "dir /4 &PCFILE";

*********************************************************************;
* Process the dir /4 command to get the date/time that the file     *;
* was created or last updated.                                      *;
*********************************************************************;
data _null_;

format filetime time5. filedate date9.;

length timec $6  fdate $10;

infile extfile pad missover length=length;  /* For virtual drives */
```

```
  /* Code to determine PC file name - used to get right line of piped DOS command
output */

   pcfilenm = trim(left("&PCFILE"));
   pcfilenm = reverse(pcfilenm);
   r = index(pcfilenm,'\');
   substr(pcfilenm,r,length(pcfilenm) - r + 1) = ' ';
   pcfilenm = trim(left(reverse(pcfilenm)));

  /* End of code to determine PC file name */

input @;

input bigline $varying200. length;

   /* Only accept line with date/time on it */
if index(bigline,trim(left(pcfilenm))) < 1 then delete;

   /* Process the Date */
       fdate = scan(bigline,4,' ');
       filedate =
mdy(input(substr(fdate,1,2),2.),input(substr(fdate,4,2),2.),input(substr(fdate,7,4),4.
));

  /* Process the Time */
       timec=left(trim(scan(bigline,5,' ')));
       filetime = hms(input(substr(timec,1,(index(timec,':')
                    -1)),2.),input(substr(timec,(index(timec,':')+1),2),2.),01);
       if index(timec,'p') > 0 and substr(timec,1,2) ne '12' then
              filetime = filetime + 43200; /* Adjust time for PM */

  /* Create the Macro variables */
       call symput('FILETIME',put(filetime,time5.));
       call symput('FILEDATE',put(filedate,date9.));

run;

*********************************************************************;
* Deallocate the EXTFILE fileref.                                   *;
*********************************************************************;
filename extfile clear;

%MEND PCFLDTTM;
```

Process_Rtrace.sas

```
*********************************************************************************;
* Program: Process_Rtrace.sas                                                   *;
*                                                                               *;
* Purpose: This program reads an RTRACE file. It removes SAS CORE, TEMP, and the*;
*          initial GRAPH entries. It reports which SAS modules were used, and   *;
*          updates a production file.                                           *;
*                                                                               *;
*          This program is in the form of a SAS Macro and accepts two           *;
*          parameters:                                                          *;
*                                                                               *;
*             PRODFILE - This is the full path filename of the SAS data library  *;
*                        where the cumulative SAS data set RTRACE.sas7bdat will  *;
*                        reside                                                  *;
*             RTRACEFL - This is the full path filename of the RTRACE flat file  *;
*                        that will be processed by this program.                 *;
**********************************************************************************;

%MACRO RTRACERD(PRODFILE,RTRACEFL);
*********************************************************************;
* Allocate the RTRACE flat file.                                    *;
*********************************************************************;
filename rtracefl "&RTRACEFL";

*********************************************************************;
* Determine if the RTRACE flat file is in use.                      *;
*********************************************************************;
data _null_;

inuse = fopen('rtracefl');

call symput('INUSE',inuse);

if inuse = 0 then do;

      put '*** Attention: the file below was in use ***';
      put "&RTRACEFL";
      put _all_;
      put '*** Attention: the file above was in use ***';

end;

run;

%IF &INUSE NE 0 %THEN %DO;
*********************************************************************;
* Allocate the RTRACE production SAS data set.                      *;
*********************************************************************;
libname  prodfile "&PRODFILE";

*********************************************************************;
* Get the PC Date/Time - These are created at file close time.      *;
*********************************************************************;
%PCFLDTTM(&RTRACEFL);

*********************************************************************;
* Determine the SYSTEM ID and create the SYSTEM SAS Macro variable.*;
*********************************************************************;
data _null_;

call symput('SYSTEM',substr("&RTRACEFL",3,3));

run;

*********************************************************************;
* Process the records in the RTRACE flat file. Specific selection  *;
* criteria weed out all but the special SAS product .DLL files.    *;
*********************************************************************;
```

```
data tracetmp(keep= system userid strtdate strttime enddate endtime product);

length userid $15 strtdate enddate 8 strttime endtime 8  rtracefl $70  system $3;

length product $8;

retain system "&SYSTEM";
retain userid  strtdate  strttime enddate endtime;
retain firstrec 0;

label system    = 'System ID'
         userid    = 'User ID'
      strtdate  = 'Start Date'
      strttime  = 'Start Time'
        enddate = 'End Date'
        endtime = 'End Time'
      product   = 'SAS Product'
      ;

format strtdate enddate worddate.  strttime endtime time8.;

infile rtracefl length=lenvar;

input @;

input bigline $varying200. lenvar;

        /*****************************************************/
        /* Delete unneeded entries.                          */
        /*****************************************************/
if index(bigline,'F:\sas82\SAS') < 1 and
   index(bigline,'C:\Program Files\SAS Institute\SAS\V8\') < 1 then delete;
if index(bigline,'.DLL') < 1 then delete;
if index(bigline,'F:\sas82\SAS\graph\sasexe\sasxglob.DLL') > 0 then delete;


        /******************************************************/
        /* Decompose entries that are left to get the PRODUCT.*/
        /******************************************************/
if index(bigline,'F:\sas82\SAS') then do;
       x = index(bigline,'F:');
       bigline = substr(bigline,x,lenvar-x+1);
       y = index(substr(bigline,14,36),'\');
       product = substr(bigline,14,y-1);
end;
else if index(bigline,'C:\Program Files\SAS Institute\SAS\V8') then do;
       x = index(bigline,'C:');
       bigline = substr(bigline,x,lenvar-x+1);
       y = index(substr(bigline,39,36),'\');
       product = substr(bigline,39,y-1);
end;

        /*****************************************************/
        /* Get Userid, Date, time.                           */
        /*****************************************************/
if firstrec = 0 then do; /* FIRSTREC is a flag that the first viable record was
processed */
```

```
   rtracefl = trim(left("&RTRACEFL"));
   rtracefl = reverse(rtracefl);
   r = index(rtracefl,'\');
   substr(rtracefl,r,length(rtracefl) - r + 1) = ' ';
   rtracefl = trim(left(reverse(rtracefl)));

     /* Obtain point of reference to decomp record */
   n = index(rtracefl,'.rtd');
   userid = substr(rtracefl,1,n-16);      /* Get Userid */

   rtracefl = substr(rtracefl,n-15,15);   /* Get start Date   */
   strtdate = input(substr(rtracefl,1,9),date9.);

   rtracefl = substr(rtracefl,10,6);     /* Get start Time   */
   strttime = hms(input(substr(rtracefl,1,2),2.),
                 input(substr(rtracefl,3,2),2.),input(substr(rtracefl,5,2),2.));

   enddate = "&FILEDATE"d;               /* Get End Date from PC date */
   endtime = "&FILETIME"t;               /* Get End Time from PC time */

   firstrec = 1; /* Set flag that first record has been processed */

end;

run;

*********************************************************;
* Determine if there are any obs in TRACETMP.           *;
*********************************************************;
proc sql noprint;
select nobs - delobs into :numbrobs
       from dictionary.tables
        where libname = "WORK" and
              memname = "TRACETMP"
              ;
quit;

  /*********************************************************/
  /* Only do the following if there are obs in TRACETMP. */
  /*********************************************************/
%IF &NUMBROBS NE 0 %THEN %DO;

********************************************************************;
* Sort the data by product/fullpath/action.                        *;
********************************************************************;
proc sort data=tracetmp nodupkey;
        by strtdate strttime system userid product;
run;

*************************************************************;
* Update the RTRACE production SAS data set. If the data set*;
* exists update it, if this is the first time, allocate it. *;
*************************************************************;
%IF %SYSFUNC(exist(prodfile.rtrace)) %THEN %DO;
       data prodfile.rtrace;
       modify prodfile.rtrace tracetmp;
```

```
              by strtdate strttime system userid product;

              if _iorc_= 0 then replace;
              else output;

       run;
%END;
%ELSE %DO;
       data prodfile.rtrace;
       set  tracetmp;
        by strtdate strttime system userid product;
       run;
%END;

  /********************************************************/
  /* End of DO stmt. to process obs in TRACETMP.          */
  /********************************************************/
%END;
%ELSE %DO;
*********************************************************************;
* Make note in log that no valid products were found in the        *;
* RTRACE file.                                                      *;
*********************************************************************;
data _null_;

       put '*** Attention: No product records were found in this file ***';
       put "RTRACE File: &RTRACEFL";
       put '*** Attention: No product records were found in this file ***';

run;

  /********************************************************/
  /* End of DO stmt. to flag no valid products in RTRACE.*/
  /********************************************************/
%END;

*********************************************************************;
* Delete the RTRACE flat file.                                      *;
*********************************************************************;
data _null_;

       rc = fdelete('rtracefl');

       put '*********************************************';
       put 'FILE DELETE RETURN CODE = ' rc;
       put '*********************************************';

run;

  /********************************************************/
  /* End of DO stmt. to process RTRACE files not in use. */
  /********************************************************/
%END;

*********************************************************************;
* Deallocate the RTRACE fileref.                                    *;
*********************************************************************;
```

```
filename rtracefl clear;

********************************************************************;
* Deallocate the RTRACE production SAS data set.                   *;
********************************************************************;
libname prodfile clear;

%MEND RTRACERD;
```

Rtrace_Reports.sas

```
*******************************************************************************;
* Program: Rtrace_Reports.sas                                                 *;
*                                                                             *;
* Purpose: This program creates a number of reports based on RTRACE data.    *;
*******************************************************************************;

%LET FROMDATE = '01MAR2004'd;
%LET   TODATE = '31MAR2004'd;

*****************************************************************;
* Allocate the RTRACE SAS data library.                         *;
*****************************************************************;
libname rtrace 'Q:\data';

*****************************************************************;
* Subset RTRACE production data set to get report date range.   *;
*****************************************************************;
data rtrace / view=rtrace;
set rtrace.rtrace(where=(strtdate between &FROMDATE and &TODATE));

if _n_ = 1 then do;
   call symput('REPTFROM',trim(left(put(&FROMDATE,worddate.))));
   call symput('REPTTO',trim(left(put(&TODATE,worddate.))));
end;

format elapsed time5.;
label  elapsed = 'Total Session Elapsed Time (HH:MM)';

elapsed = endtime - strttime;
if elapsed <= '00:01't then elapsed = '00:01't;

run;

*********************************************************************;
* Create a report of overall SAS product Usage for the entire month.*;
*********************************************************************;
proc summary nway data=rtrace;
     class product userid;
     var elapsed;
output out=sumprod(drop=_type_ elapsed rename=(_freq_ = sessions)) sum=;
run;

proc summary nway data=sumprod;
     class product;
     var sessions;
```

```
output out=summc(drop=_type_ rename=(_freq_ = users)) sum=;
run;
proc print data=summc noobs label;
label sessions = 'Number of Times Product Was Used'
      users    = 'Number of individual Users';
var product users sessions;
title1 'Number of Times Each SAS Product Was Used';
title2 'On Corporate Windows Servers';
title3 "For &REPTFROM Through &REPTTO";
run;

*********************************************************************;
* Report of days/# of distinct SAS users.                           *;
*********************************************************************;
proc summary nway data=rtrace(where=(upcase(product)='CORE'));
       class strtdate userid;
       var   elapsed;
output out=sumusery(drop=_type_ _freq_) sum=;
run;

proc summary nway data=sumusery;
       class userid;
       var   elapsed;
output out=sumuserz(drop=_type_ rename=(_freq_ = sessions)) sum=;
run;

proc summary nway data=sumuserz;
       class sessions;
       var   elapsed;
output out=sumuserw(drop=_type_ rename=(_freq_ = users)) sum=;
run;

proc sort;
       by descending sessions;
run;

proc print data=sumuserw noobs label;
label users    = 'Number of Individuals Using SAS This Many Days'
      sessions = 'Number of Days SAS Was Used' ;
var sessions Users;
sum          Users;
title1 'Number of Distinct SAS Users';
title2 'On Corporate Windows Servers';
title3 'Summarized by Number of Days SAS Was used';
title4 "For &REPTFROM Through &REPTTO";
run;

*********************************************************************;
* Create a report of overall SAS usage for individual user by day.  *;
*********************************************************************;
proc summary nway data=rtrace(where=(upcase(product)='CORE'));
       class strtdate userid;
       var   elapsed;
output out=sumusers(drop=_type_ rename=(_freq_ = sessions)) sum=;
run;

proc summary nway data=sumusers;
```

```
       class strtdate;
       var   sessions;
output out=sumuser2(drop=_type_) sum=;
run;

proc print data=sumuser2 noobs label;
label _freq_ = 'Number of Individual SAS Users'
      sessions = 'Total Number of SAS Sessions';
var strtdate _freq_ sessions;
sum                 sessions;
title1 'Number of Distinct SAS Users';
title2 'On Corporate Windows Servers';
title3 'Summarized by Date';
title4 "For &REPTFROM Through &REPTTO";
run;

*********************************************************************;
* Create a report of SAS product Usage by distinct day.             *;
*********************************************************************;
proc summary nway data=rtrace;
     class strtdate product userid;
     var elapsed;
output out=sumprod(drop=_type_ elapsed rename=(_freq_ = sessions)) sum=;
run;

proc summary nway data=sumprod;
     class strtdate product;
     var sessions;
output out=sumpro2(drop=_type_ rename=(_freq_ = users)) sum=;
run;


proc print data=sumpro2 noobs label;
label sessions = 'Number of Times Product Was Used'
      users    = 'Number of Individual Users';
       by strtdate;
       id strtdate;
var product users sessions;
title1 'Number of Times Each SAS Product Was Used';
title2 'On Corporate Windows Servers';
title3 "For &REPTFROM Through &REPTTO";
run;
```

Appendix B. – Contents Listing of the Windows Servers RTRACE SAS Data Set

```
 -----Alphabetic List of Variables and Attributes-----

#    Variable    Type    Len    Pos    Format       Label
---------------------------------------------------------------
3    enddate     Num       8      8    WORDDATE.    End Date
5    endtime     Num       8     24    TIME8.       End Time
6    product     Char     15     47                 SAS Product
2    strtdate    Num       8      0    WORDDATE.    Start Date
4    strttime    Num       8     16    TIME8.       Start Time
7    system      Char      3     62                 System ID
1    userid      Char     15     32                 User ID
```

Appendix C. – Examples of RTRACE-based SAS Usage Reports

This Appendix contains some examples of RTRACE-based SAS usage reports. Because of SUGI paper size restrictions, only a portion of each report is included. Also, to protect the privacy of the author's organization, the numbers in the reports have been altered.

```
            Number of Times Each SAS Product Was Used
                    On Corporate Windows Servers
          For December 1, 2003 Through December 31, 2003

                                         Number of
                          Number of        Times
               SAS       individual       Product
             Product        Users         Was Used

             access             5              29
             af                 1               9
             connect            5              36
             core             191            6551
             ets                1               5
             fsp                3               9
             graph              5              62
             iml                2               8
             or                 1               6
             qc                 1              11
             stat               5              59

                   Number of Distinct SAS Users
                    On Corporate Windows Servers
            Summarized by Number of Days SAS Was used
          For December 1, 2003 Through December 31, 2003

                                   Number of
                                  Individuals
                    Number of      Using SAS
                     Days SAS      This Many
                     Was Used        Days

                        31             1
                        30             1
                        24             1
                        23             1
                        22             5
                        21             6
                        20            12
                        19             8
                         .             .
                         .             .

                                  ===========
                                      191
```

```
                Number of Distinct SAS Users
              On Corporate Windows Servers
                  Summarized by Date
    For December 1, 2003 Through December 31, 2003

                                          Total
                           Number of    Number of
                          Individual       SAS
        Start Date         SAS Users     Sessions

      December 1, 2003        184          1436
      December 2, 2003        100           750
      December 3, 2003        191          1721
      December 4, 2003         50           689
      December 5, 2003        132          1042
      December 6, 2003          7            14
       .      .      .            .            .
       .      .      .            .            .

      Number of Times Each SAS Product Was Used
              On Corporate Windows Servers
    For December 1, 2003 Through December 31, 2003

                                             Number of
                                Number of      Times
                      SAS      Individual     Product
  Start Date        Product       Users      Was Used

December 1, 2003    access           2             6
                    af               3             5
                    connect          5            70
                    core           184          1436
                    ets              3             3
                    fsp              2             4
                    graph            2             8
                    iml              1             1
                    or               1             5
                    qc               1             3
                    stat             4            71
                     .               .             .
```