

Paper 465-2013

## Knowing Me, Knowing My UI (ah-haa): Understanding SAS® 9.3 and SAS® 9.4 Desktop and Web Client Application Usage within Your Organization

Edoardo Riva and Simon Williams, SAS Institute Inc., Cary, NC

### ABSTRACT

Understanding which groups of users are using which sets of desktop and web applications can help your organization increase efficiency while reducing the risk to your operations. This paper details the importance of understanding users and desktop application interactions, and the methods by which these interactions can be defined and reported on. Working examples using SAS® desktop and web applications such as SAS® Enterprise Guide®, SAS® Data Integration Studio, SAS® Management Console, and SAS® Web Report Studio are presented.

### INTRODUCTION

For many organizations, having traceability for members of staff accessing and leveraging a solution that is based on the SAS Intelligence Platform, whether through a desktop or web-based client, is a key requirement. Going one step further, there might be a requirement to track users' access by the type of SAS client application the users are leveraging.

There is the obvious security focus when leadership teams think about traceability, but there are additional reasons as well:

- Understanding which users are likely to be affected by updates and upgrades to software. During times when customer sites are looking to apply hotfixes or to update or migrate to a new version of SAS, the IT teams would benefit from knowing which users are affected and could prioritize their efforts accordingly.
- IT teams that are in charge of packaging and provisioning processes for desktop clients can take a more targeted approach for building packages based on known usage by the groups of users.
- Using traceability to identify potentially broken work practices and processes. From an organization's perspective, it can help to understand whether users are using the right tools for the specific activities and actions taken based on the results of the audit.
- Key processes and supporting applications can be identified from a business perspective (for example, during business process reviews).
- Traceability can help the organization implement any monitoring required to ensure that licensing requirements are being met.

### BASIC LEVEL OF MONITORING

Almost every IT team has two basic lists: client applications and end users. They most likely have a table to match which users are enabled to use a particular application. There are dozens of tools to support the automatic discovery and inventory of software installed on desktops and laptops, but these do not necessarily provide all the information required.

When SAS software evolved from being stand-alone software installed on analysts' desktops (SAS Display Manager System or SAS® Foundation) to an enterprise platform with a multi-tier architecture, it gained the automatic logging of all user activity on the server side. The functionality to determine which users have been logging in to the SAS Intelligence Platform has been available for SAS® 9.1.3, SAS® 9.2, and SAS® 9.3. As the SAS Intelligence Platform continues to evolve, introducing both desktop and web-based new clients, so will the functionality to log the different types of users and client application usage.

Figure 1 illustrates the SAS Web Administration Console that helps to track end users currently active on SAS web client applications.

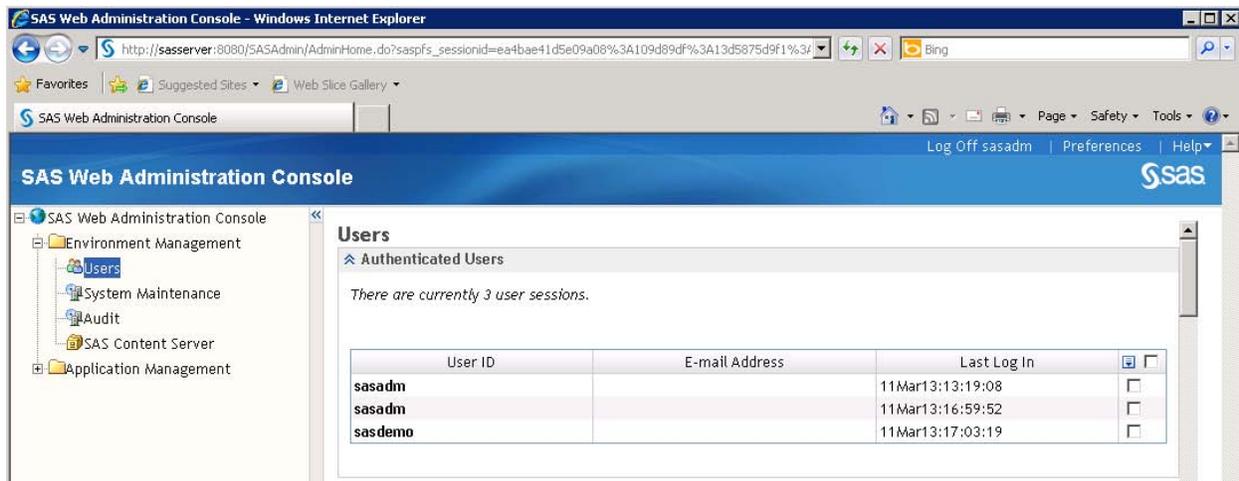


Figure 1 - SAS Web Administration Console, Active Users List

## ADVANCED LEVEL OF MONITORING

What happens when there is a critical fix that involves only a given client application used by a limited number of developers, or a project with a tight timeline requires to upgrade the client of a specific business solution? It might not be possible to wait on scheduled software maintenance windows, or to upgrade all software on all desktops. This is just a simple scenario describing why more advanced reporting is required; we need to answer questions that usually distill down to this:

### *Which users are using which SAS applications?*

In the rest of the paper, the authors will present information and methods to track the usage of the SAS client software. The scope of the work included the following applications:

#### Desktop applications

- SAS Enterprise Guide
- SAS Data Integration Studio
- SAS Management Console

#### Web applications

- SAS Web Report Studio
- SAS® BI Dashboard
- SAS® Information Delivery Portal
- SAS Stored Process Web Application
- SAS Web Administration Console
- SAS® Forecast Studio (SAS Forecast Studio is a full desktop client, but it works in conjunction with a web application deployed on the middle tier that leverages the same auditing and logging frameworks as any other web application.)

## USER AND CLIENT APPLICATION USAGE AUDITING PATH

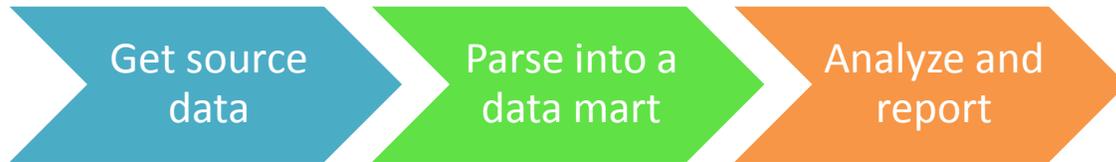
There are different ways to determine which users have been accessing the range of client applications. Elements of the default configuration of the SAS Intelligence Platform provide the opportunity for some level of auditing. To increase the level of auditing requires additional configuration of the SAS Intelligence Platform, in particular the enabling of and analysis of server-side logging, and the creation of some additional reports to be defined.

Determining the proper source of information is critical because it might be scattered in multiple places (logs, auditing

tables, metadata, and so on).

Proper planning and adopting a consolidated methodology is important in order to avoid the risk of missing some piece of information or of having to consult a lot of different sources.

We propose a simple methodology that builds on three points:



1. Find the appropriate source  
How do we know that an end user is using a client application? Through the “tracks” client applications leave within SAS log files, data, or metadata. This might require changing the logging configuration of some applications and server-side processes, or just verifying that the expected information is present in the right place. Usually, this requires the execution of tests for every client application, so it might be necessary to ask the collaboration of end users to participate in the testing.
2. Parse logs in to a data mart  
Having identified all the sources, it is quite difficult to derive meaningful information from them without consolidating them in a data mart. Like every data flow in your enterprise, the “simple” task requires you to extract data from the sources, transform it, and load it in a set of structured tables... hey, that’s ETL. SAS can do it!
3. Build a set of consistent reports – or define tools for free analysis  
There is a big difference between having all the data we need and getting meaningful information out of it. Only a good set of reports or the right analysis tool can help us answer the business questions that started our quest. Again, with SAS we have all the tools we need.

While we are building this “reporting framework”, we are required to consider the following:

- different behavior between client applications – some write their information in server logs, some in audit tables, some applications currently do not write enough of the information we require.
- risk of building a new custom application that is difficult to maintain, with the need to change it when we deal with new platforms, software solutions, or SAS releases.
- need to deploy an auditing process that will have minimum impact on the existing environment. For example, excessive logging can slow down the processing time of end-user applications.

To get the desired outcome while minimizing the above risks, we can use a set of tools already available for your SAS environment.

## TOOLS FOR BUILDING THE USER AND CLIENT APPLICATION USAGE AUDITING PATH

When preparing to undertake the auditing, understanding which tools are required is key. The following tools are required:

- the audit framework
- the logging framework
- data loading and reporting jobs
- The SAS® Audit, Performance and Measurement Package

We present details of each of the tools below.

### AUDIT FRAMEWORK

SAS web applications and other SAS middle-tier services provide auditing features.

Depending on the application and its configuration, these auditing features can record all actions performed both by the direct users of the system and by the system itself.

Some applications might provide a more complete audit, detailing not only the actions that are performed but also the states of the objects that are affected by those actions.

The key feature is that logon, logoff, and unsuccessful logon attempts create audit records for all SAS web applications. Beginning with SAS 9.3, these audit records are stored within the SAS® Web Infrastructure Platform Database in the relational tables SAS\_AUDIT and SAS\_AUDIT\_ENTRY.

**Important Note.** The *SAS Intelligence Platform Middle-Tier Administration Guide* warns against direct access to the data within the audit tables. The authors can be contacted to obtain further information about this subject.

We require very basic information from that database, and we can rely on the above mentioned Administration Guide to make sure that all audit records contain the following information:

- user ID that performed the audited action.
- action that occurred. This is stored as an action code.
- data and time that the audited action occurred.

The documentation contains a table to interpret the action codes we require:

**Table 5.4** Common Audit Object Types and Actions

Audit Action	Object Type ID Value	Action Type ID Value
User log on	-1	8
Use log off	-1	9

**Figure 2 - Audit Table Documentation**

Accessing the SAS\_AUDIT tables, we can see some very useful data, such as that shown in Figure 3.

	AUDIT_ID	USER_ID	ACTION_TYPE_ID	TIMESTAMP_DTT	AUDIT_INFO	SUCCESS	SESSION_ID	EXE
32	115	sasadm	8	06MAR13:10:38:10		1	ea4bae41d5e09a08:109d89df:13d38613f26:1b1f	
33	201	sasdemo	8	06MAR13:12:41:47	Information Delivery Portal 4.3	1	ea4bae41d5e09a08:109d89df:13d38613f26:2ee0	
34	202	sasdemo	9	06MAR13:12:43:22		1	ea4bae41d5e09a08:109d89df:13d38613f26:2ee0	
35	203	sasinst	8	06MAR13:12:54:27		0	None - Failed Logon	
36	204	etluser	8	06MAR13:12:56:55		1	ea4bae41d5e09a08:109d89df:13d38613f26:3103	
37	205	etluser	9	06MAR13:13:00:15		1	ea4bae41d5e09a08:109d89df:13d38613f26:3103	
38	206	etluser	8	06MAR13:13:02:17		1	ea4bae41d5e09a08:109d89df:13d38613f26:6416	
39	207	etluser	9	06MAR13:13:34:49		1	ea4bae41d5e09a08:109d89df:13d38613f26:6416	
40	208	etluser	8	06MAR13:16:06:53		1	ea4bae41d5e09a08:109d89df:13d38613f26:7cd4	
41	209	etluser	9	06MAR13:16:45:31		1	ea4bae41d5e09a08:109d89df:13d38613f26:7cd4	
42	210	sasdemo	8	07MAR13:00:34:52		0	None - Failed Logon	
43	211	sasdemo	8	07MAR13:00:35:10	Web Report Studio 4.3	1	ea4bae41d5e09a08:109d89df:13d418a7403:52d8	
44	212	sasdemo	9	07MAR13:00:41:56		1	ea4bae41d5e09a08:109d89df:13d418a7403:52d8	
45	213	sasadm	8	07MAR13:07:01:19	Web Administration Console 9.3	1	ea4bae41d5e09a08:109d89df:13d418a7403:35fa	
46	214	sasdemo	8	07MAR13:08:43:15	Web Report Studio 4.3	1	ea4bae41d5e09a08:109d89df:13d418a7403:216f	
47	301	etluser	8	08MAR13:09:41:40		1	ea4bae41d5e09a08:109d89df:13d418a7403:53b2	
48	302	etluser	9	08MAR13:09:46:17		1	ea4bae41d5e09a08:109d89df:13d418a7403:53b2	
49	303	sasdemo	8	08MAR13:09:47:47	Stored Process Web App 9.3	1	ea4bae41d5e09a08:109d89df:13d418a7403:5f7e	
50	304	sasdemo	9	08MAR13:09:50:00		1	ea4bae41d5e09a08:109d89df:13d418a7403:5f7e	
51	305	sasdemo	8	08MAR13:10:03:28	Web Report Studio 4.3	1	ea4bae41d5e09a08:109d89df:13d418a7403:617f	
52	306	sasdemo	9	08MAR13:10:08:43		1	ea4bae41d5e09a08:109d89df:13d418a7403:617f	

**Figure 3 - SAS\_AUDIT Table Content from a SAS 9.3 Environment**

It is important to consider that not all logon (type=8) records have an associated application, and that more applications might be related to a single logon. This is because in SAS 9.3 logins are more global than product accesses. So you cannot assume that a login session belongs to a single product or that only one product is accessed during its lifetime. In order to verify this information, you will need to do the following:

1. Select all records with action=8 (logon) and success. (This gives you the set of all successful login sessions.)

2. For each record from this group, select all records whose SESSION\_ID matches that in the login record.
3. Select all unique non-zero-length values of AUDIT\_INFO or EXECUTOR\_NM from this last set to produce the set of products accessed during that session.

For some products, generally you will have a 1-to-1 relationship between products and sessions, largely because of how rich clients operate. But for many web applications though, you'll see a many-to-1 relationship. A method for implementing the three-step approach is provided within the "Additional Web Applications" in the Samples section below.

## LOGGING

The SAS 9.3 logging facility is a flexible, configurable framework that you can use to collect, categorize, and filter events, and write them to a variety of output devices. The logging facility supports problem diagnosis and resolution, performance and capacity management, and auditing and regulatory compliance.

The logging facility is used by most SAS server processes. You can also use the logging facility within SAS programs.

The logging facility provides detailed information, which provides the opportunity to track access to almost every server process within the SAS Intelligence Platform. Some of the default settings already provide a good level of information—for example, the MetadataServer log tracks every logon and logoff activity, with details that include user name, client IP address, date, and time information. A key fact, which client software is connecting to the server, is present only for some clients.

You must change the default logging for some servers to get, wherever it is possible, additional information without causing additional performance drops, in order to be able to track all the desired client applications.

We will not describe the full syntax of the logging framework—you will find the official guide in the paper references—but it will ease the reader to understand two key elements we will be leveraging to fulfill our needs: *loggers* and *appenders*.

A logger is a named entity that enables you to track log events whose name corresponds to the logger. In simple terms, it tells what to log.

An appender is an entity that specifies the destination for the log message. It also specifies how the message is formatted and provides additional filtering capabilities. Again, to make it simple, it tells where and how to log what comes from loggers.

In the Samples section, we explain how to make use of loggers and appenders in order to get the desired logging of client application interaction.

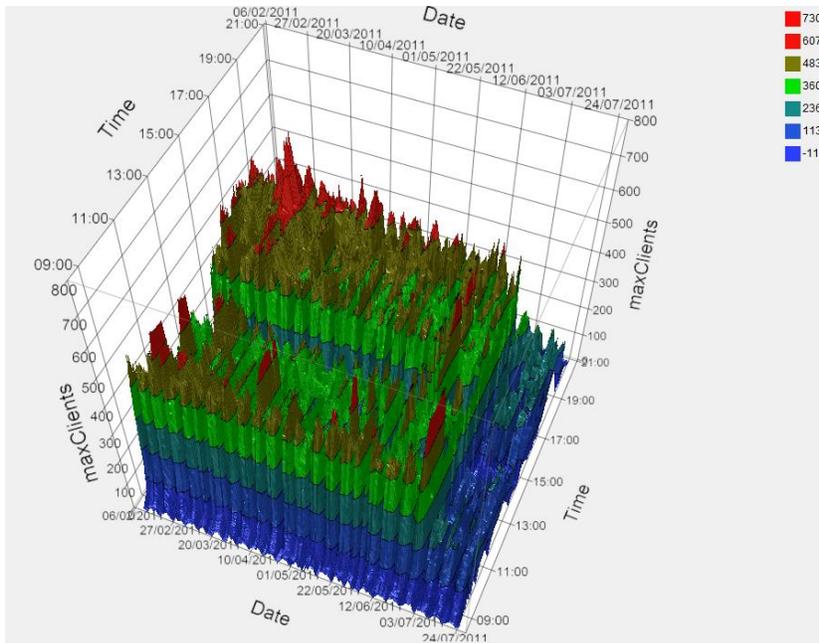
## DATA LOADING AND REPORTING JOBS

It is always possible to write custom SAS jobs to parse and report on any type of data, whether it comes from the audit framework, from logging, or from any other source.

The authors have written a set of macros to accomplish the following tasks:

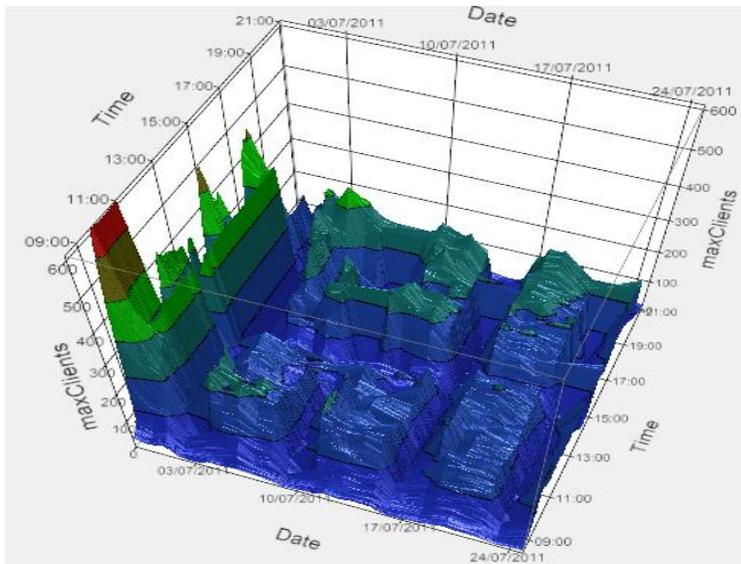
- parse a directory to get a list of the IOM log files inside it
- for every selected file, read its content line by line into a data set
- join the lines corresponding to the same user session, calculating some statistics like the total number of connected clients per minute

Once the data was inside a table, the use of SAS analysis software produced graphs to show the number of client connections. See figures 4 and 5 for examples.



**Figure 4 - Clients Connected to the Metadata Server**

In figure 4, the organization used the analysis to understand whether there were too many client connections (more than 500 concurrent connections each minute of the day in working hours for the whole 2011, when it was expected to be less than half of this). Figure 5 was produced after a solution was implemented within the organization. It clearly shows that the solution reduced the maximum number of concurrent client to below 200.



**Figure 5 - Clients Connected to the Metadata Server (Detail)**

Unless there is a specific need, and as long as it is simple enough to easily get the desired outcome, we suggest that you leverage an already existing framework, or, at least integrate the custom code into it. We think that the framework that best fits the scope of this paper is the SAS Audit, Performance and Measurement Package.

### THE SAS AUDIT, PERFORMANCE AND MEASUREMENT PACKAGE

The SAS Audit, Performance and Measurement Package is a collection of utilities provided by SAS that customers can download, install, and configure for their organization. The package enables a site using the SAS® Enterprise BI

suite to monitor the state of the SAS 9.3 BI architecture, implement audit authentication reports for regulatory compliance, and report on the performance and usability of the SAS®9 BI analytic server environment.

The features and capabilities of the package are enabled through a set of SAS language programs, libraries, and operating-system-specific scripts. A site can easily extend these features by creating custom reports to meet local site reporting or audit requirements.

The SAS Audit, Performance and Measurement Package fits most of our requirements thanks to the information organized and stored inside its data mart. For example, the `Artifactusagedetails` table contains a comprehensive list of objects (or artifacts), actions performed on them, timestamps, and user IDs of the actual users. From that it is easy to get a report of end users who are using some applications like SAS Web Report Studio or SAS Stored Process Web Application.

Even though this paper does not cover the details of how to install and configure this package, we will leverage it to show how it streamlines our processes and places them in the path of a well-designed framework. See the Samples section below for more details.

## SAMPLES

Following are some practical examples of how to track client usage with some of the desktop and web applications, with the methods and tools previously outlined.

## WEB APPLICATIONS

### SAS Web Report Studio

For customers who use SAS Web Report Studio 4.3 under SAS 9.3, the auditing functionality is provided out of the box. There is no need to make any elaborate configuration changes; it just works right away. It is very easy to implement auditing with SAS Web Report Studio. A detailed example is provided below. For more information about this topic (which is referred to as "Key User Action"), read the [SAS® 9.3 Intelligence Platform: Web Application Administration Guide](#). The documentation provides descriptions of the key user action codes (there are currently 13) and how to set up the roll-over attributes of the key user logs.

There are 3 steps required to quickly take a look inside the logs and see the information related to audits and events:

1. Identify the location and names of the relevant log files are kept:
  - By default, the directory location is `...\config\Lev1\Web\Logos`.
  - The archive logs have the default name format `SASWebReportStudio4.3_KeyActions.log_yyyymmdd_hrrmss` (for example, `SASWebReportStudio4.3_KeyActions.log_20130129_221231`).
  - The live or current log is named `SASWebReportStudio4.3_KeyActions.log`.
2. To minimize the risk of losing the original log files, you should make a copy of the original log file.
  - Copy `SASWebReportStudio4.3_KeyActions.log_20130129_221231`, and rename the copy to `SASWebReportStudio4.3_KeyActions31Jan2013.XML`.
3. Open a SAS Display Manager System session and submit the relevant code to accomplish the following tasks:
  - Use a `LIBNAME` statement to point to the XML-based data in the log.
  - Create an XML table called `Event` (via `proc datasets`).
  - Create a SAS data set to use in further processing.

With the SAS data set created, it is then possible to view the relevant information or perform further actions. For true implementations, the expectation is to have the access and reporting on key user actions built into an automated process using proven data integration and business intelligence practices.

**Important Note.** If you must access a log file before it has been archived (that is, the log file is in active use), then ensure that after you have copied the files, you edit the copy to add the closing tag `</wrskeyuseractions>` to the bottom of the log file. If you do not add the tag, you will receive an error message when running the following `LIBNAME` statement:

```
<?xml version="1.0" encoding="UTF-8" ?>
|<wrskeyuseractions>
<event><javaDate>1359582595076</javaDate><date>1/30/13</date>
<event><javaDate>1359582608127</javaDate><date>1/30/13</date>
```

Figure 6 - The Header Information within the Key User Actions Log Files

```
<![CDATA[Stuart]]></user><session>4</session><sasSession>5fec51256554cfa0:21ff6be1:13c8d5fbb53:4271</sasSession><code>0</
<![CDATA[Stuart]]></user><session>4</session><code>2</code><description>Logoff</description></event>
<![CDATA[Rob]]></user><session>5</session><sasSession>5fec51256554cfa0:21ff6be1:13c8d5fbb53:42b3</sasSession><code>0</cod
<![CDATA[Rob]]></user><session>5</session><code>2</code><description>Logoff</description></event>
<![CDATA[Erwan]]></user><session>6</session><sasSession>5fec51256554cfa0:21ff6be1:13c8d5fbb53:42ec</sasSession><code>0</c
<![CDATA[Erwan]]></user><session>6</session><code>2</code><description>Logoff</description></event>
><![CDATA[sasdemo]]></user><session>7</session><sasSession>5fec51256554cfa0:21ff6be1:13c9123cb88:-776f</sasSession><code>
><![CDATA[sasdemo]]></user><session>7</session><sasSession>5fec51256554cfa0:21ff6be1:13c9123cb88:-776f</sasSession><code>
><![CDATA[sasdemo]]></user><session>7</session><code>2</code><description>Logoff</description></event>
><![CDATA[Stuart]]></user><session>8</session><sasSession>5fec51256554cfa0:21ff6be1:13c9123cb88:-61df</sasSession><code>0
><![CDATA[Stuart]]></user><session>8</session><code>3</code><description>Save</description><report>/Shared Data/SAS Testi
><![CDATA[Stuart]]></user><session>8</session><code>2</code><description>Logoff</description></event>
```

Figure 7 - A Snippet of the Content within the Key User Action Log Files

The SAS code used to read the Key User Action log file is very simple:

```
libname myxml2 xml
'C:\sas93\config\Lev1\Web\Logs\SASWebReportStudio4.3_KeyActions31Jan2013Midday.XML'
;
proc datasets library=myxml2;
quit;

data wrsaudit31012013;
set myxml2.Event;
run;
```

An example of the type of data within the Key User Action log files after it has been read into a SAS data set is provided in Figure 9.

REPORT	SASSESSION	SESSION	USER	DESCRIPTION	CODE	TIME	DATE
		.		System Startup	9	16:49:00	2013-01-30
	5fec51256554cfa0:21ff6be1:13c8d5fb	1	sastrust@saspw	Logon	0	16:50:00	2013-01-30
	5fec51256554cfa0:21ff6be1:13c8d5fb	2	sastrust@saspw	Logon	0	16:50:00	2013-01-30
		2	sastrust@saspw	Logoff	2	16:50:00	2013-01-30
		1	sastrust@saspw	Logoff	2	22:07:00	2013-01-30
	5fec51256554cfa0:21ff6be1:13c8d5fb	3	sastrust@saspw	Logon	0	22:07:00	2013-01-30
	5fec51256554cfa0:21ff6be1:13c8d5fb	4	Stuart	Logon	0	05:40:00	2013-01-31
		4	Stuart	Logoff	2	05:47:00	2013-01-31
	5fec51256554cfa0:21ff6be1:13c8d5fb	5	Rob	Logon	0	05:47:00	2013-01-31
		5	Rob	Logoff	2	05:47:00	2013-01-31
	5fec51256554cfa0:21ff6be1:13c8d5fb	6	Erwan	Logon	0	05:48:00	2013-01-31
		6	Erwan	Logoff	2	05:48:00	2013-01-31
	5fec51256554cfa0:21ff6be1:13c9123	7	sasdemo	Logon	0	10:42:00	2013-01-31
	5fec51256554cfa0:21ff6be1:13c9123	7	sasdemo	Logon	0	10:42:00	2013-01-31
		7	sasdemo	Logoff	2	10:45:00	2013-01-31
	5fec51256554cfa0:21ff6be1:13c9123	8	Stuart	Logon	0	12:16:00	2013-01-31
/Shared Data/SAS Testing/Test Reports/ProdSaleWRSReport		8	Stuart	Save	3	12:19:00	2013-01-31
		8	Stuart	Logoff	2	12:19:00	2013-01-31
	5fec51256554cfa0:21ff6be1:13c9123	9	Rob	Logon	0	12:20:00	2013-01-31
/Shared Data/SAS Testing/Test Reports/CLASS_WRSReport		9	Rob	Save	3	12:23:00	2013-01-31
		9	Rob	Logoff	2	12:23:00	2013-01-31
	5fec51256554cfa0:21ff6be1:13c9123	10	Erwan	Logon	0	12:25:00	2013-01-31
/Shared Data/SAS Testing/Test Reports/ProdSaleWRSReport		10	Erwan	Open	4	12:25:00	2013-01-31
/Shared Data/SAS Testing/Test Reports/ProdSaleWRSReport1		10	Erwan	Save	3	12:26:00	2013-01-31
		10	Erwan	Logoff	2	12:26:00	2013-01-31

**Figure 8 - A Snippet of the SAS Data Set that is Created after Reading the Key User Action Log File**

From this, it is possible to determine the following events:

- the SASTRUST account being used to log on to SAS Web Report Studio at system start up time
- The users Stuart, Rob, and Erwan all logged on and logged off to SAS Web Report Studio during the course of the working day
- Stuart saved a report ProdSaleWRSReport (and because Stuart only saved the report and did not open it, you might be able to infer he is actually the report creator)
- Rob saved a report CLASS\_WRSReport (and because Rob only saved the report and did not open it, you might be able to infer he is actually the report creator)
- Erwan opened the ProdSaleWRSReport report and saved it as ProdSaleWRSReport1

### Additional Web Applications

All of the web applications that track user activities in the audit framework can be tracked and reported using custom code to access the data within the SAS Web Infrastructure Platform Database . The custom code has been tested on different web applications, including SAS Web Report Studio, SAS BI Dashboard, SAS Information Delivery Portal, SAS Forecast Studio, SAS Stored Process Web Application, and SAS Web Administration Console.

The authors created the wipSessions.sas program and placed it into the SAS Audit, Performance and Measurement Package Jobs directory, which is:

```
... \APM93 \SASEnvironment \SASCode \Jobs
```

This was then added within the SAS Audit, Performance and Measurement Package master program runArtifactAnalyzer.sas by adding the following line at the end:

```
%inc "SASEnvironment\slash.SASCode\slash.Jobs\slash.wipSessions.sas" / source2;
```

Here is some of the code of the program:

```
data wipAudit;
  set wip.sas_audit;
```

```

run;

Proc SQL;
/* select logon rows */
create table wipLogon as
select timestamp_DTTM as dtm_logon, user_id, session_id,
       "Other" as description
from wipAudit
where action_type_id=8
;
/* join with logoff rows */
create table wipSession as
select wipLogon.*, wipAudit.timestamp_DTTM as dtm_logoff
from wipLogon left Join wipAudit on (wipLogon.session_id=wipAudit.session_id
and (wipAudit.action_type_id=9 or wipAudit.action_type_id is null))
;
/*find description*/
create table wipProd as
select distinct session_id, case
when audit_info is null then executor_nm
else audit_info
end as description
from wipAudit where audit_info is not null or executor_nm is not null
;
quit;
/* update description */
data wipSession;
modify wipSession wipProd;
by session_id;
run;

/* copy table in staging library */
proc copy in=work out=&outlib.;
select wipSession;
run;

```

### Code 1 - wipSessions.sas

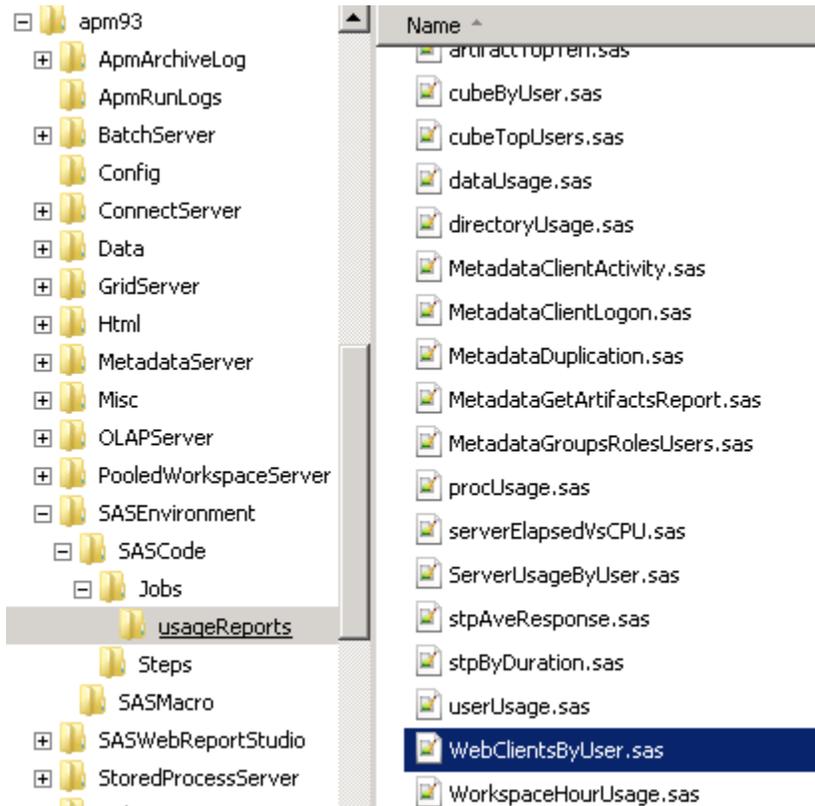
The result of running the code creates a table similar to the one in Figure 10, which contains all the required information to track usage of web client applications by user and day/time.

VIEWTABLE: Artifact.Wipsession					
	dtm_logon	USER_ID	SESSION_ID	description	dtm_logoff
23	06MAR2013:12:41:46.5050	sasdemo	ea4bae41d5e09a08:109d89df:13d38613f26:2ee0	Information Delivery Portal 4.3	06MAR2013:12:43:21.9680
24	06MAR2013:12:56:54.7570	etluser	ea4bae41d5e09a08:109d89df:13d38613f26:3103	Other	06MAR2013:13:00:14.5820
25	06MAR2013:13:02:16.5740	etluser	ea4bae41d5e09a08:109d89df:13d38613f26:6416	Other	06MAR2013:13:34:48.5730
26	06MAR2013:16:06:53.2750	etluser	ea4bae41d5e09a08:109d89df:13d38613f26:7cd4	Other	06MAR2013:16:45:31.4870
27	07MAR2013:08:43:15.4520	sasdemo	ea4bae41d5e09a08:109d89df:13d418a7403:216f	Web Report Studio 4.3	.
28	07MAR2013:07:01:18.7770	sasadm	ea4bae41d5e09a08:109d89df:13d418a7403:35fa	Web Administration Console 9.3	.
29	07MAR2013:00:35:09.7740	sasdemo	ea4bae41d5e09a08:109d89df:13d418a7403:52d	Web Report Studio 4.3	07MAR2013:00:41:56.1120
30	08MAR2013:09:41:40.0090	etluser	ea4bae41d5e09a08:109d89df:13d418a7403:53b2	Other	08MAR2013:09:46:16.9510
31	08MAR2013:09:47:47.4570	sasdemo	ea4bae41d5e09a08:109d89df:13d418a7403:5f7e	Stored Process Web App 9.3	08MAR2013:09:50:00.3350
32	08MAR2013:10:03:28.4650	sasdemo	ea4bae41d5e09a08:109d89df:13d418a7403:617f	Web Report Studio 4.3	08MAR2013:10:08:43.2650
33	08MAR2013:12:17:15.0740	sasadm	ea4bae41d5e09a08:109d89df:13d418a7403:7842	Other	.
34	08MAR2013:12:54:34.2680	sasdemo	ea4bae41d5e09a08:109d89df:13d418a7403:7ae1	SAS Forecast Server	08MAR2013:12:59:43.1120
35	09MAR2013:10:27:14.1610	sasadm	ea4bae41d5e09a08:109d89df:13d4b511d47:292	Other	.
36	11MAR2013:11:22:49.4840	etluser	ea4bae41d5e09a08:109d89df:13d5875d9f1:342e	BI Dashboard 4.3	.
37	11MAR2013:11:17:30.2130	sasdemo	ea4bae41d5e09a08:109d89df:13d5875d9f1:3673	Information Delivery Portal 4.3	11MAR2013:11:19:21.0000
38	11MAR2013:10:30:53.7010	sasdemo	ea4bae41d5e09a08:109d89df:13d5875d9f1:4747	Information Delivery Portal 4.3	.

Figure 9 - wipSessions Table

When the table is under the SAS Audit, Performance and Measurement Package staging area, it is possible to write a new report using the facilities of the package. The new report creates a summary table, providing a list of counts by web application for each SAS user. (See Figure 12.)

To accomplish this, a new program was put in the ...\APM93\SASEnvironment\SASCode\Jobs\usageReports directory. The SAS Audit, Performance and Measurement Package is set up by default to run any program found in that directory.



**Figure 10 - The SAS Audit, Performance and Measurement Package usageReports Directory**

To maximize effectiveness, the new program uses the same set of macros and the same structure as the ones already present. Figure 12 provides a succinct summary of the users and the number of times they have accessed a specific SAS web client application.

**Web Client Software usage by Users for 1 Week(s)**

**The MEANS Procedure**

USER_ID	description	N Obs
etuser	BI Dashboard 4.3	1
	Other	4
sasadm	Other	3
	Web Administration Console 9.3	1
sasdemo	Information Delivery Portal 4.3	6
	Other	2
	SAS Forecast Server	1
sasinst	Stored Process Web App 9.3	1
	Web Report Studio 4.3	3
	Other	1

**Figure 11 - Web Client Users Report**

## DESKTOP APPLICATIONS

There are different methods for determining desktop client application usage. For client applications like SAS Enterprise Guide and SAS Data Integration Studio, the authors leveraged output in the SAS Workspace Server log. This was done by first modifying the logging settings in order to ensure that key information for tracking client connections was being generated.

By default, the SAS Audit, Performance and Measurement Package is set up and configured with SAS Workspace Server logging switched off for performance considerations. This is required to be enabled, but the authors made use of filtering to limit the amount of logging being produced.

To enable logging, append a `-logconfigloc` option to the SAS Workspace Server `sasv9_usermods.cfg` file in order to enable that server's supplied `logconfig.apm.xml` file. By default the files are located in the following directory:

```
<SAS-configuration-directory>\Levl\SASApp\WorkspaceServer
```

The new `sasv9_usermods.cfg` will contain the following line:

```
-logconfigloc "C:\SAS93\Config\Levl\SASApp\WorkspaceServer\logconfig.apm.xml"
```

The next step is to add the required loggers and appenders to the `logconfig.apm.xml` file (always remember to make a backup copy of the original file. If you place any syntax error in the `logconfig` file, the SAS Workspace Server will not start anymore). The new logger will read:

```
<!-- Client access log -->
<appender name="ClientAccess" class="FileAppender">
  <param name="Append" value="false"/>
  <param name="Unique" value="true"/>
  <param name="ImmediateFlush" value="true"/>
  <param name="FileNamePattern"
value="C:\SAS93\Config\Levl\SASApp\WorkspaceServer\Logs\ClientAccess_%S{username}_%
d{yyyy-MM-dd_HH.mm.ss}_%S{hostname}_%S{pid}.log"/>
  <layout>
    <param name="ConversionPattern" value="%d %-5p [%t] %c:%X{Client.ID}:%u -
%m"/>
  </layout>
  <filter class="StringMatchFilter">
    <param name="StringToMatch" value="SAS Enterprise Guide"/>
    <param name="AcceptOnMatch" value="true"/>
  </filter>
  <filter class="StringMatchFilter">
    <param name="StringToMatch" value="SAS Data Integration Studio"/>
    <param name="AcceptOnMatch" value="true"/>
  </filter>
  <filter class="StringMatchFilter">
    <param name="StringToMatch" value="LaunchedIOMServer"/>
    <param name="AcceptOnMatch" value="true"/>
  </filter>
  <filter class="DenyAllFilter"></filter>
</appender>
```

The key is to filter the lines that contain the client software name (first two filters) or the open and close of the server session (third filter), and then to deny writing anything else (last filter). We are sending this information to a new log file, which we will later parse.

The actual logging is produced by the logger elements we add at the bottom of the same file:

```

<!-- Application message logger -->
  <logger name="App.Program" additivity="false">
    <level value="info"/>
    <appender-ref ref="ClientAccess"/>
  </logger>
<logger name="App.LaunchedIOMServer.LaunchedIOMServer" additivity="false">
  <level value="debug"/>
  <appender-ref ref="ClientAccess"/>
</logger>

```

One key option to include on all additional logger definitions is `additivity="false"`. This option ensures that the log messages are sent only to the new appender. Without this option, the messages go to both the default appenders and to the new appender.

There is no need to restart any server; every new connection will start a SAS Workspace Server session what will pick up the new logging configuration. The new log contains information about session start and close, and lists the client name for SAS Enterprise Guide and SAS Data Integration Studio, as shown in Output 1.

```

2013-03-11T12:48:23,299 DEBUG [00000117]
App.LaunchedIOMServer.LaunchedIOMServer.SetActivity:3:sasdemo@SGCWIN02114 - IOM CALL
{compRef:3494960}->LaunchedIOMServer::SetActivity()
2013-03-11T12:48:23,300 DEBUG [00000117]
App.LaunchedIOMServer.LaunchedIOMServer.SetActivity:3:sasdemo@SGCWIN02114 - IOM
RETURN 0={compRef:3494960}->LaunchedIOMServer::SetActivity()
2013-03-11T12:48:25,964 INFO [00000013] App.Program::sasdemo@SGCWIN02114 - 7
%LET _CLIENTAPP = 'SAS Enterprise Guide';
2013-03-11T12:48:25,972 INFO [00000007] App.Program::sasdemo@SGCWIN02114 -
20130311:12.48.25.96: 00000003: :: IOM LOG: 7 %LET _CLIENTAPP =
'SAS Enterprise Guide';
2013-03-11T16:01:55,695 DEBUG [00006333]
App.LaunchedIOMServer.LaunchedIOMServer.Close::sasdemo@SGCWIN02114 - IOM CALL
{compRef:3494960}->LaunchedIOMServer::Close()
2013-03-11T16:01:55,696 DEBUG [00006333]
App.LaunchedIOMServer.LaunchedIOMServer.Close::sasdemo@SGCWIN02114 - IOM RETURN
0={compRef:3494960}->LaunchedIOMServer::Close()

```

#### Output 1 – SAS Workspace Server Log for SAS Enterprise Guide Connections

After we have the desired information in our logs, we can use a SAS program to parse it; we prefer to integrate it with the SAS Audit, Performance and Measurement Package like we did for the web applications because the package provides macros and samples to parse these types of logs. You can find samples of the code online on the [sascommunity.org](http://www.sascommunity.org/wiki/Knowing_me,_knowing_my_UI_(ah-haa):_Understanding_SAS%C2%AE_9.3_and_SAS%C2%AE_9.4_Desktop_and_Web_Client_Application_Usage_Within_Your_Organization) at [http://www.sascommunity.org/wiki/Knowing\\_me,\\_knowing\\_my\\_UI\\_\(ah-haa\):\\_Understanding\\_SAS%C2%AE\\_9.3\\_and\\_SAS%C2%AE\\_9.4\\_Desktop\\_and\\_Web\\_Client\\_Application\\_Usage\\_Within\\_Your\\_Organization](http://www.sascommunity.org/wiki/Knowing_me,_knowing_my_UI_(ah-haa):_Understanding_SAS%C2%AE_9.3_and_SAS%C2%AE_9.4_Desktop_and_Web_Client_Application_Usage_Within_Your_Organization).

## THE FUTURE IS NOW: SAS 9.4

Starting with SAS 9.3, it was possible for a client application (for example, a desktop application or a web application) to pass its name to a SAS IOM server. The server can use the name for logging and auditing.

The new default behavior is for the name of your application to appear in IOM server logs whenever your application causes a connection to be made to an IOM server. For example, when SAS Web Report Studio causes a connection to a workspace server, it results in messages similar to Output 2:

```

2013-02-03T14:42:30,177 INFO [00000756] :etluser - New client connection (16)
accepted from server port 63583 for SAS token user etluser. Encryption level is
Credentials using encryption algorithm SASPROPRIETARY. Peer IP address and port are
[:ffff:10.96.9.148]:57780 for APPNAME=Web Report Studio 4.3.

```

#### Output 2 - Sample 9.3 Object Spawner Log

Because this was new a new feature in SAS 9.3, it is used only in a select number of client applications.

In SAS 9.4, almost all client applications pass their name to the servers, so it will be a lot easier to track them!

For example, in SAS 9.3 there is no way to tell if an access to the SAS Metadata Server belongs to SAS Management Console or to other clients like SAS® Information Map Studio or SAS® OLAP Cube Studio, just to name the most used.

In SAS 9.4, there is a clear distinction. Even with the default log, you can find SAS Management Console entries from the user (in this case sasadm@saspw) as shown in Output 3:

```
2013-03-07T12:44:29,258 INFO [00042176] :sasadm@saspw - New client connection
(2242) accepted from server port 8561 for user sasadm@saspw. Encryption level is
Credentials using encryption algorithm SASPROPRIETARY. Peer IP address and port are
[::ffff:10.96.5.24]:61898 for APPNAME=SASManagementConsole 904000.
```

**Output 3 - Sample SAS 9.4 SAS Metadata Server Object Log Showing a Connection via the SAS Management Console**

Similarly, in Output 4, it is possible to view a connection to the SAS Metadata Server being made through the Information Delivery Portal web application by the sasdmo user.

```
2013-03-11T20:18:46,498 INFO [00107229] :sasdmo - New client connection (6116)
accepted from server port 8561 for SAS token user sasdmo. Encryption level is
Credentials using encryption algorithm SASPROPRIETARY. Peer IP address and port are
[::ffff:10.96.3.58]:55095 for APPNAME=Information Delivery Portal 4.4.
```

**Output 4 - Sample SAS 9.4 SAS Metadata Server Object Log Showing a Connection via the SAS Information Delivery Portal**

By default, the SAS Audit, Performance and Measurement Package tracks these logs and parses the APPNAME parameter/value pairs. The package *artifacts* and *audit\_transactions* SAS data sets contain a wealth of data that will drive insightful reports.

## CONCLUSION

Starting with SAS 9.3, the ability to understand the usage of SAS client applications by an organization's user community became a reality. Developments that included the auditing tables within the SAS Web Infrastructure Platform Database and the inclusion of the APPNAME parameter within SAS logs have provided the required source data on which to provide insightful reports. With the advent of SAS 9.4, the amount of data that can be used to power such reporting has increased. Combined with other data, there has never been a better time to understand how the user community within an organization leverages The Power to Know with SAS and the ability for the IT team to provide enhanced support for the SAS user community.

## REFERENCES

- SAS Institute Inc. 2012. "Audit, Performance and Measurement Package for SAS® 9.3 Enterprise Business Intelligence". Available as part of the Audit, Performance and Measurement Package Package at <http://support.sas.com/rnd/emi/APM93/index.html>.
- SAS Institute Inc.(2011. "SAS® 9.3 Logging: Configuration and Programming Reference." Available at <http://support.sas.com/documentation/cdl/en/logug/63131/PDF/default/logug.pdf>.
- SAS Institute Inc. 2011. "SAS® 9.3 Interface to Application Response Measurement (ARM): Reference." Available at <http://support.sas.com/documentation/cdl/en/armref/62980/PDF/default/armref.pdf>.
- SAS Institute Inc. 2012. "SAS® 9.3 Intelligence Platform: Web Application Administration Guide, Third Edition." Available at <http://support.sas.com/documentation/cdl/en/biwaag/65230/PDF/default/biwaag.pdf>.
- SAS Institute Inc. 2012. "SAS® 9.3 Intelligence Platform: Middle-Tier Administration Guide, Third Edition." <http://support.sas.com/documentation/cdl/en/bimtag/65708/PDF/default/bimtag.pdf>.

## ACKNOWLEDGEMENTS

The authors would like to thank colleagues within the Global Architecture and Technology Enablement (GATE) team for their support in writing this paper. In addition, the authors would like to thank the R&D teams for their support with answering our questions.

## CONTACT INFORMATION

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

Edoardo Riva  
SAS Institute Inc.  
SAS Campus Drive  
E-mail: [edoardo.riva@sas.com](mailto:edoardo.riva@sas.com)

Simon Williams  
SAS Institute Inc.  
SAS Campus Drive  
E-mail: [simon.williams@sas.com](mailto:simon.williams@sas.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.