

Auditing an Enterprise SAS® Visual Analytics 6.2 Environment with SAS® Tools: From the SAS® IT Perspective

Dan Lucas, Brandon Kirk, SAS Institute Inc., Cary, NC

ABSTRACT

With a growing enterprise analytics environment comprised of global users and a variety of sensitive data sources, a system administrator is faced with several challenges. These include knowing who logs in to the system, how often, and what applications and data sources are being consumed. This information is necessary for auditing the consumers of data as well as monitoring the growth of data sources for hardware expansion. With the usage of the SAS® Audit, Performance and Measurement Package, along with some additional middle-tier logging and SAS code, information about the major consumers of the environment can be loaded into LASR tables and analyzed with SAS Visual Analytics reporting tools.

INTRODUCTION

With SAS Visual Analytics, business users, statisticians, and data scientists have the ability to work with large amounts of data to design reports, use powerful analytics, and deliver results to the web or mobile devices. With the quest for more data for analysis, an administrator is faced with more questions around security and auditing of the various data sources and system usage. IT management and compliance groups often ask the following types of questions:

- Who is logging in to the system, and which applications are they using?
- Who is accessing the LASR tables, how often, and what application are they using?
- How many LASR tables exist in the system, and how fast are they growing?
- What are the peak usage times?
- Who has Read access privileges for secured data sources?

This paper covers some basic usage of the SAS Audit, Performance and Measurement (APM) Package to help resolve some of these questions. In addition, enhancements can be added to the default APM package to garner more details that are specific to a SAS Visual Analytics environment. The additional data assists with system usage, system growth, and data access concerns. The level of reporting that is provided from these areas provides answers for IT management of current usage and future system planning. Because the framework for data collection and reporting is based on SAS, system administrators with SAS programming skills can augment the tools to extract more site-specific details. This discussion focuses on the following main areas:

- SAS Audit, Performance and Measurement Package setup and reports
- middle-tier logging for SAS Visual Analytics applications
- metadata queries

Each of the following sections provides some examples of data collection and reporting capabilities and provides a framework to help foster additional enhancements that are needed by each site.

SAS 9.4 AUDIT, PERFORMANCE AND MEASUREMENT PACKAGE

The SAS Audit, Performance and Measurement package (APM) for SAS® 9.4 Enterprise Business Intelligence (EBI) is a collection of scripts and SAS programs that are provided by SAS that customers can download, install, and customize for their organizations. You can review and download the APM package from the <http://support.sas.com/rnd/emi/APM94/index.html> web site. The APM package allows an EBI site to monitor the state of the environment, implement auditing of the environment, and report on performance of the EBI components. The components of APM can be modified to create and deploy a framework to meet operational requirements for each IT group. The APM package relies on the reading and parsing of various log files that are generated in an enterprise environment.

Because SAS Visual Analytics cannot license all of the components of EBI, you can customize the APM package to focus on the data that is extracted from the SAS Metadata Server, a subset of other BI servers, and their respective log files. The following section covers system logins and metadata object permission changes. This information can be collected and reported from a default installation of the APM package.

SAS ENVIRONMENT MANAGER OPERATIONS INTELLIGENCE BUNDLE

The SAS Environment Manager Operations Intelligence bundle extends the real-time monitoring and metric collection capabilities of SAS Environment Manager Web Application. These extended capabilities provide an event-centric approach to delivering information that empowers SAS administrators with information to understand the operational health of their SAS environments. In addition, the Operations Intelligence bundle provides a starting point for further analysis: tying anomalies to specific events, log transactions, user activities within the SAS environment, performing reboot cause analysis. Included with the bundle are:

- Log collectors and parsers (APM)
- Log Collector/Log Distributor Web Services
- Datamart ETL
- Report Center
- Event importing and exporting
- Solution Kit Framework

The final set of components in SAS Environment Manager bundle is a framework for extending SAS Environment Manager to support SAS solutions and technologies. This framework provides a flexible mechanism for developers to integrate their solution with SAS Environment Manager monitoring environment.

This paper covers the modifications of the SAS Audit, Performance and Measurement package to enable the collection of SAS Visual Analytics usage information. These modifications were used with Visual Analytics 6.1, 6.2 and 6.3 installations. With the evolving SAS Environment Manager Operations Intelligence bundle, these same modifications will be converted to a Visual Analytics “solutions kit”. These future enhancements will require less modifications to the standard SAS Environment Manager Operations Intelligence bundle, will utilize the Log Collector Web Services, and will integrate with the SAS Environment Manager datamart.

QUICK START GUIDE TO SAS AUDIT, PERFORMANCE AND MEASUREMENT CONFIGURATION

The first step in monitoring and auditing a SAS Visual Analytics environment is the installation of the SAS Audit, Performance and Measurement package for SAS® 9.4. Eventually, this installation will evolve to the SAS Environment Manager Operations Intelligent bundle and configuration of a solutions kit. You can install and set up APM with a quick start configuration. The following steps are a summary of the details that are covered in the installation guide, *Audit, Performance and Measurement Package for SAS 9.4 Enterprise Business Intelligence*. This section does not cover all of the details of APM installation and configuration. For more information, see the installation guide, which is located at “Downloads & Hot Fixes: SAS Audit, Performance and Measurement” (<http://support.sas.com/demosdownloads/setupcat.jsp?cat=SAS+Audit%2C+Performance+and+Measurement>).

Create an apm94 Directory

The APM installation process involves the creation of an **apm94** directory and extraction of a TAR file that contains various scripts and SAS programs. You can simplify the APM environment by creating this directory on a server that can access SAS Metadata Server log files and SAS Web Application Server middle-tier log files. These log files are transferred from their default locations to staging directories in the **apm94** directory. Output 1 shows the creation of **apm94** and the extraction of its components.

```
[node00] /opt/mis/sas> ls ./tmp/*.tar
./tmp/APM94.unx.tar
[node00] /opt/mis/sas> mkdir apm94
[node00] /opt/mis/sas> cd apm94
[node00] /opt/mis/sas/apm94> tar xf ../tmp/APM94.unx.tar
[node00] /opt/mis/sas/apm94> ls -l
ApmArchiveLog
ApmRunLogs
apm_status.sh
autoexec.sas
```

```

BatchServer
Config
configureAPM_autoexec.sas
configureAPMlocal.sh
configureAPM.sas
configureAPM.sh
ConnectServer
Data
GridServer
Html
MetadataServer
Misc
OLAPServer
PooledWorkspaceServer
readSASLogs.sh
revision_history.txt
runCheckServer.sh
runCheckSetinit.sh
runDataArchive.sh
runSASReports.sh
runStatusInit.sh
runStatusUpdate.sh
SASEnvironment
SASWebReportStudio
setEnvLogs.cfg.sh
StoredProcessServer
Utilities
WorkspaceServer

```

Output 1. Output from Creation of apm94

Modify apm94 Environment Files

After you have created the **apm94** directory, complete the following steps:

1. Modify the setenvLogs.cfg.sh file and update and review the lines that are identified by "DEFINE_THIS_VALUE" or "REVIEW_THIS_VALUE". These entries must match your specific environment.
2. Modify configureAPM_autoexec.sas to point to the SAS Metadata Server for your environment.
3. Execute the configureAPM.sh script that uses these configuration files and generates another local script that is executed on each of the hosts in the environment. Output 2 provides an example of executing these next two scripts.

```

[node00] /opt/mis/sas/apm94> ./configureAPM.sh
Enter sasadm@saspw password (not persisted)
XXXXXXX

Review /opt/mis/sas/apm94/ApmRunLogs/configureAPM.sas.log

[node00] /opt/mis/sas/apm94> ./configureAPMlocal_node00.sh

```

Output 2. Output from configureAPM.sh and configureAPMlocal_node00.sh

Disable ARM Logs for EBI Servers

For a SAS Visual Analytics environment, you might not be concerned with the logging of application response measurement (ARM) for the various EBI Servers (SAS Pooled Workspace Server, SAS Stored Process Server, and so on). The ARM logs are enabled as part of the APM setup scripts. To disable these extra logs:

1. In the **SASConfig/SASApp/** directory, comment out the "%apm_armsetup;" invocation that was added to the appserver_autoexec_usermods.sas program.
2. For each server under the **SASApp** directory, comment out the -LOGCONFIGLOC option that was added to

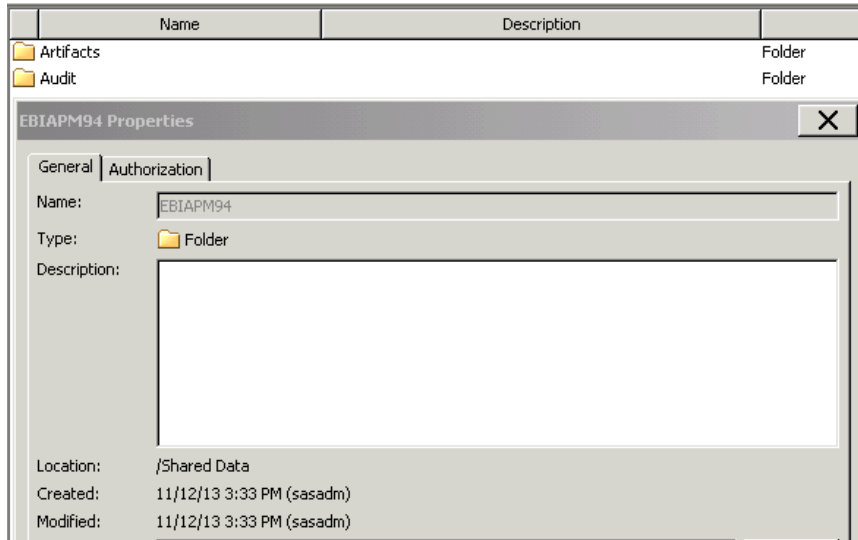
sasv9_usermods.cfg. This option references a logconfig.apm.xml file.

By disabling the ARM setup of the EBI servers, the main component that is being monitored is the activity through the metadata server.

After the updates of these environment files and execution of their scripts, restart your metadata server and BI server to create the additional log files that are processed by the APM package.

Import APM Metadata Definition

One of the final steps for APM setup is importing a package of components that are registered in the metadata server. In the `./Utilities/` directory, execute the `spkImport.sh` script to import a stored process and register several tables and folders in the metadata location `/Shared Data/EBIAPM94`. Display 1 is an example of the `/Shared Data/EBIAPM94` folder that is created by the import script.



Display 1. APM Package `/Shared Data/EBIAPM94` Metadata Folder

When you expand the `/Shared Data/EBIAPM94/Artifacts` folder, you see the various datasets that accumulate information from the log collection and parsing process. Display 2 shows the members of the **Artifacts** subfolder that are used by the reporting tools.

Name	Description	Type	Last Modified
AUDIT_ACCESS		Table	Nov 12, 2013 3:33:44 PM
AUDIT_ADUSER		Table	Nov 12, 2013 3:33:44 PM
AUDIT_GROUP		Table	Nov 12, 2013 3:33:44 PM
AUDIT_TRANSACTIONS		Table	Nov 12, 2013 3:33:44 PM
AUDITACCESSCONTROLDETAIL		Table	Nov 12, 2013 3:33:44 PM
EBIAPM Artifact		Library	Nov 12, 2013 3:33:44 PM

Display 2. APM Package `/Shared Data/EBIAPM94` Metadata Folder

The Audit_Transactions data set is the key auditing data set to load into a LASR Analytic Server for reporting from SAS Visual Analytics applications.

Schedule Log Collection

After the servers have collected log files for at least a day or two, run the `runDataArchive.sh` script. This script moves log files into the `apm94` directory, parses the log files, saves the data, archives log files, and executes various batch reports that are shipped with `apm94`. Manually execute this script for the first few days to track the return codes in the various log files that are created by the script. After the process has been verified, schedule the script file with your operating system scheduler, such as cron. Output 3 shows the manual invocation of the `runDataArchive.sh` script.

```
[node00] /opt/mis/sas/apm94> ./runDataArchive.sh
Tue Nov 12 15:43:43 EST 2013: Initialize the apm_status file
```

```

Tue Nov 12 15:43:44 EST 2013: Completed initializing the apm_status file.
NOTE: Executing Setup Program: /opt/mis/sas/apm94/Utilities/cimport_artifact.sas
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/runArtifactAnalyzer.sas for
/opt/mis/sas/apm94/setEnvLogs_SASApp.sas
NOTE: Checking for availability of SAS/GRAPH software
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/MetadataGetArtifacts.sas
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/MetadataArtifactsPhysical.sas
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/bi_audit_reports.sas
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/usageReports/artifactBusinessHours.s
as
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/usageReports/artifactShiftHours.sas
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/usageReports/artifactStats.sas
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/usageReports/artifactTopTen.sas
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/usageReports/cubeByUser.sas
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/usageReports/cubeTopUsers.sas
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/usageReports/dataUsage.sas
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/usageReports/directoryUsage.sas
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/usageReports/MetadataClientActivity.
sas
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/usageReports/MetadataClientLogon.sas
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/usageReports/MetadataDuplication.sas

NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/usageReports/MetadataGetArtifactsRep
ort.sas
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/usageReports/MetadataGroupsRolesUser
s.sas
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/usageReports/procUsage.sas
NOTE: Executing Usage Report
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/usageReports/serverElapsedVsCPU.sas
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/usageReports/ServerUsageByUser.sas
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/usageReports/stpAveResponse.sas
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/usageReports/stpByDuration.sas
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/usageReports/userUsage.sas
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/usageReports/WorkspaceHourUsage.sas
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/usageReports/WorkspaceTopMemoryUser.
sas
NOTE: Executing Usage Report
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/usageReports/WorkspaceTopUsers.sas

```

```
Tue Nov 12 15:45:43 EST 2013: Executing Setinit Check Program
Tue Nov 12 15:45:46 EST 2013: Completed execution of
/opt/mis/sas/apm94/SASEnvironment/SASCode/Jobs/runCheckSetinit.sas.
Tue Nov 12 15:45:46 EST 2013: Write the apm_status file
Tue Nov 12 15:45:46 EST 2013: Completed writing apm_status file.
Please check /opt/mis/sas/apm94/ApmRunLogs/runDataArchive.log for
runDataArchive.sh messages
```

Output 3. Output from runDataArchive.sh

For more information about all of the components that are accessed by runDataArchive.sh, see *Audit, Performance and Measurement Package for SAS 9.4 Enterprise Business Intelligence*.

DEFAULT APM LOGIN REPORTS

By default, the APM package provides some useful audit reports that are generated in HTML format and include reports of the following events:

- access control changes
- authentication errors (Access Denied or Unknown User IDs)
- user IDs that are added or removed
- groups changes

These reports are surfaced to the administrator and can be delivered via the SAS Web Server, which is a component of the SAS 9.4 environment. Display 3 shows an example of the APM report interface.

Report URL	Report Description
Audit Reports	
AccessControlChanges.html	Access Control Changes
Administrators.html	Administrators
AuthenticationError.html	Authentication Errors
GroupChanges.html	Group Changes
LoginNotAuthorized.html	Login Not Authorized
UseridsAdded.html	User ID's Added
UseridsRemoved.html	User ID's Removed
Performance Reports	
ArtifactStats.html	Artifact Usage Time Statistics
ArtifactUsageBusinessHours.html	Artifact Usage During Business Hours
ArtifactUsageShiftHours.html	Artifact Usage by Shift
CubeByUser.html	OLAP Cube Usage by User
CubeTopUsers.html	Top OLAP Cube Users
DataUsage.html	Most Heavily Used Data Sets
DirectoryUsage.html	Most Heavily Used Directories
MetadataClientActivity.html	Metadata Client Status: Active / Inactive User ID's
MetadataClientLogonUsage.html	Metadata Client Sessions: Login Details
MetadataDuplication.html	Duplicate Metadata References for a Directory
MetadataGroupsRolesUsers.html	Metadata Groups, Roles, and Users
ProcUsage.html	SAS PROC Usage
ReportUsageTopTen.html	Top 10 Report Usage
ServerElapsedVsCPU.html	Server Comparison of Elapsed vs. CPU Time
ServerUsageByUser.html	SAS Server Usage by User
STPReportAveResponse.html	Stored Process Average Response Time

Display 3. APM Report Interface for Audit Reports

The source for the various reports exists in the `apm94/SASEnvironment/SASCode/Jobs/` or `apm94/SASEnvironment/Jobs/usageReports/` directory, in the `bi_audit_reports.sas` program. Modify this program to fit the needs for your IT group. Display 4 shows an example of the Authentication Errors report with an invalid password.

Record Event	DateTime	Metadata Userid	Remote Client IP Address	Log_Line
Access denied	01AUG2013:16:18:31.552	frtest. Peer IP address and port	::ffff:10.36.150.83	2013-08-01T16:18:31.552 WARN [00972844] :sasmeta - New client connection (1055270) rejected from server port 8561 for user frtest. Peer IP address and port are [::ffff:10.36.150.83]:33271 for APPNAME=Logon Manager 9.4.
	02AUG2013:17:04:51.248	frtest. Peer IP address and port	::ffff:10.36.150.83	2013-08-02T17:04:51.248 WARN [01087237] :sasmeta - New client connection (1056270) rejected from server port 8561 for user frtest. Peer IP address and port are [::ffff:10.36.150.83]:36129 for APPNAME=Logon Manager 9.4.

Display 4. APM Audit Report for Invalid Passwords

Display 5 shows an example of information that is captured for user IDs that are added to the system.

Administrator	DateTime	Metadata Userid	Userid	Log_Line
sasadm	02DEC2013:15:35:41.277	Person	testsas@saspw	2013-12-02T15:35:41.277 INFO [01183347] 22867:sasadm - 22867:SASADM: Changed Internal Login ObjId=A54BZS2U.BD000004 for Person Name=testsas, ObjId=A54BZS2U.AO000ITX.
	02DEC2013:15:35:41.277	Person	testsas@saspw	2013-12-02T15:35:41.277 INFO [01183347] 22867:sasadm - 22867:SASADM: Changed Internal Login ObjId=A54BZS2U.BD000004 for Person Name=testsas, ObjId=A54BZS2U.AO000ITX.

Display 5. APM Audit Report for User IDs That Are Added

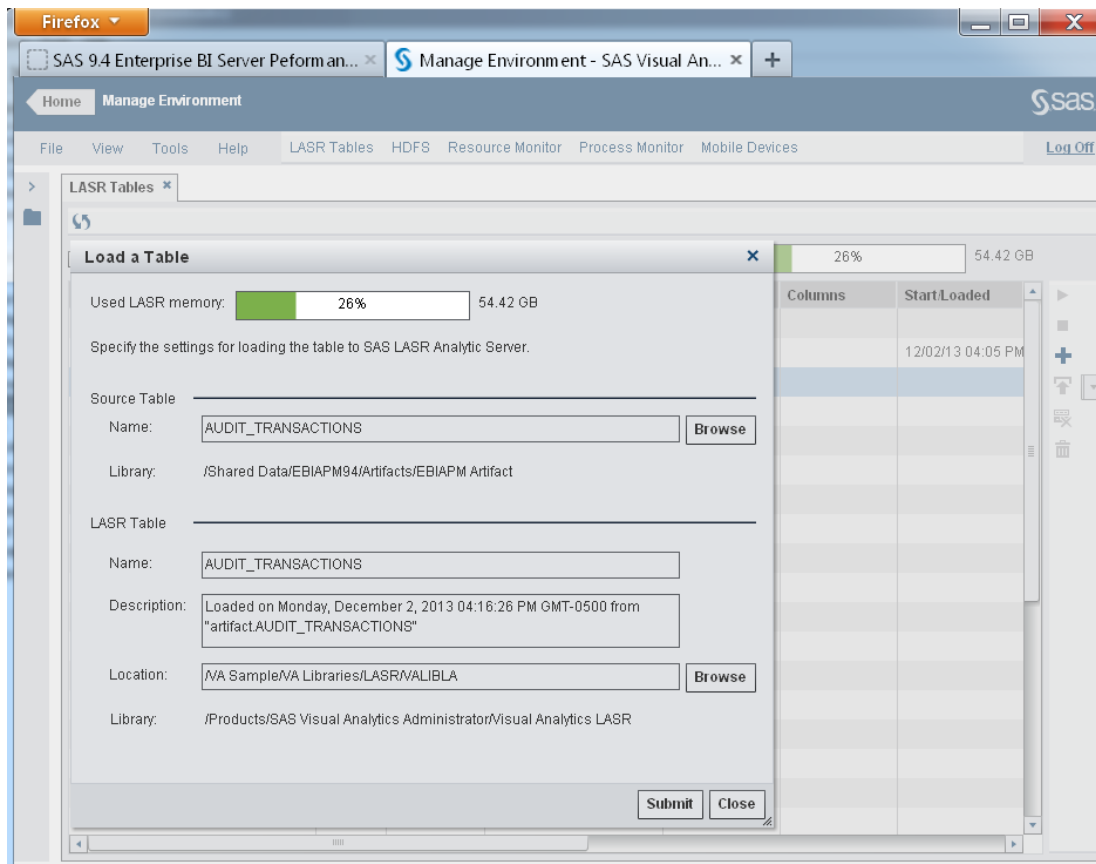
ANALYZE LOGINS WITH SAS VISUAL ANALYTICS

After the data has been loaded into a SAS data set and has been appended daily by the runDataArchive.sh script, you gain the benefit of the SAS Visual Analytics environment. Load the data set into a LASR Analytic Server where it can be interpreted with the SAS Visual Analytics suite of applications: SAS® Visual Analytics Designer, SAS® Visual Analytics Explorer, SAS® Visual Analytics Viewer, and SAS® Mobile BI. Use the **Manage Environment** tab in SAS Visual Analytics Administrator to load the Audit_Transactions table into a LASR library. Display 6 shows an example of loading a table into a LASR library.

Tables by Server Component	Status	Port	Description	Rows	Columns	Start/Loaded
LASR Analytic Server			SAS LASR Analytic server..			
Connection: LASR Analytic ...	●	10010				12/02/13 04:05 PM
Visual Analytics LASR			SAS Visual Analytics libra..			

Display 6. Load a Table into a LASR Library

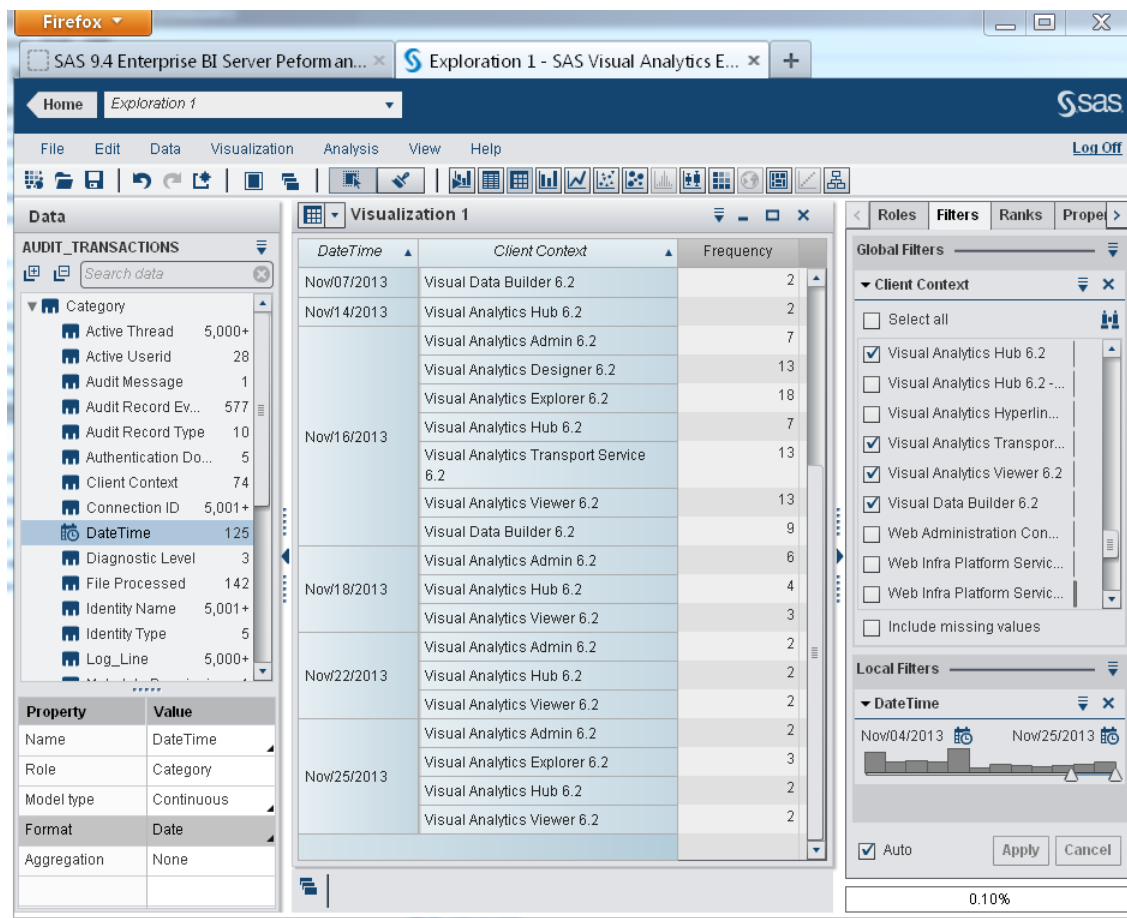
The source table information is in the location that was created during the import of the APM package using the spkImport.sh script. The LASR table information is the name of the LASR table that is created and the location is the LASR library that is used to surface the data for reporting. After this table is loaded and more data has been accumulated via the runDataArchive.sh script, drop and reload the currently loaded LASR table. Display 7 is an example of the Load a Table interface for the Audit_Transactions data set.



Display 7. Load a Table Interface for Audit_Transactions

Explore the Login Data

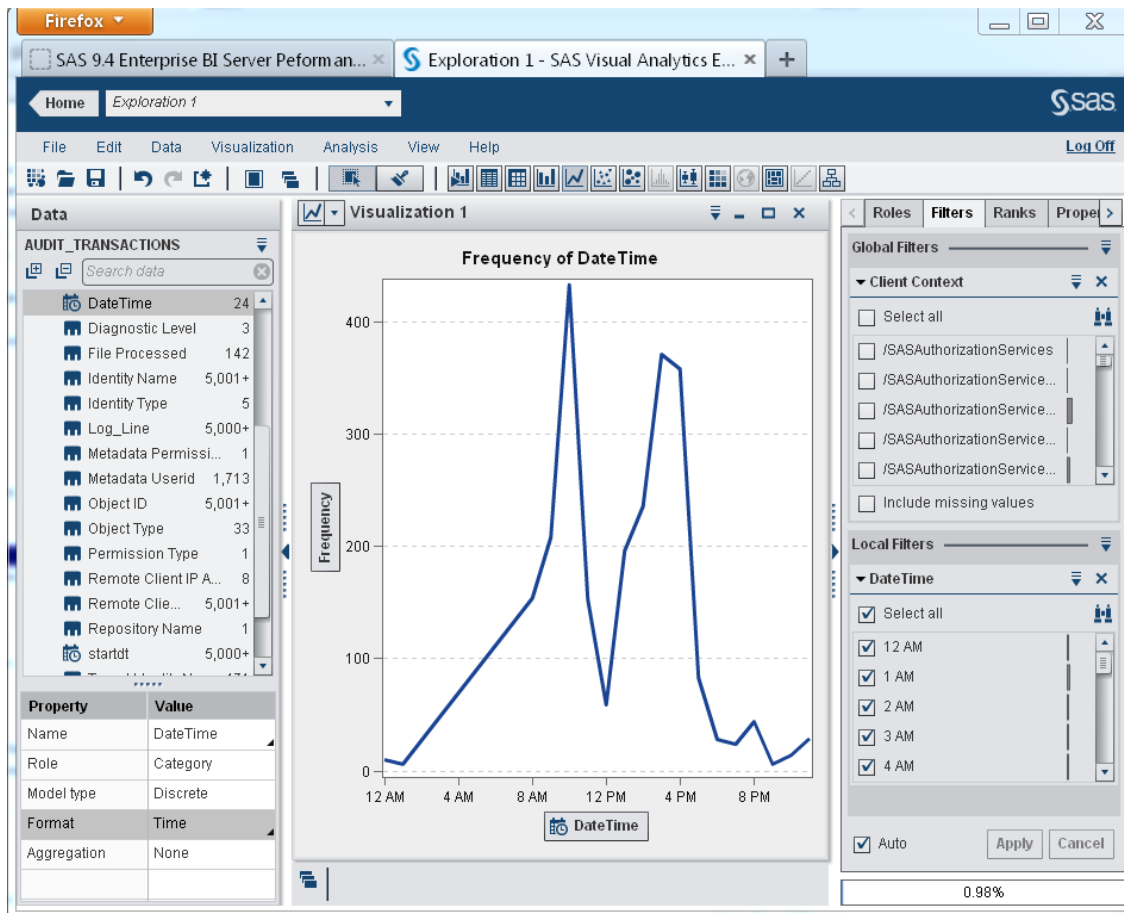
After the data has been loaded into a LASR Analytic Server and registered in metadata, you can conduct ad hoc data analysis to gain insights on login activity. Display 8 shows a simple exploration of the Audit_Transactions data set that uses a cross-tabular report to view login counts.



Display 8. Exploration of Audit_Transactions for Login Counts for Date Range

This example produces a simple count of report logins by application for a date range. The global filter was used with Client Context to limit the number of login applications. Login applications are requests for user ID validation through the metadata server. Some of the choices that are available in the global filter are: **Visual Analytics Hub 6.2**, **Visual Analytics Viewer 6.2**, **Visual Analytics Explorer 6.2**, and **Visual Analytics Designer 6.2**. The DateTime range was a local filter that limited the login times by date. This report could easily be modified by adding Active Userid to the cross tabulation to output the number of login counts by user ID.

By switching the graph output to a Line Chart, and modifying DateTime to a Time format such as HH AM and removing the usage of Client Context in the output, you can quickly see the login trend for the users of the system. Display 9 is an example of a simple login trend report.



Display 9. Login Trend Report

ENHANCE LOGIN DATA

With the SAS Metadata Server log files, the connection information that is available for a user is output with the login information (Userid) that is defined in the **Accounts** tab for the user in metadata. This shortened field might not provide the desired level of detail when building a report. To enhance this data, you can modify the `apm94/SASEnvironment/SASMacro/logauditprocesslog.sas` program to insert a step that would translate the user ID into a more common name. Execute this program for each metadata server log file that is processed. Insert the desired mapping code after the "filename auditlog;" statement, such as this example that uses an SQL view that maps user IDs to user common names.

```
/* Clear the filename before starting the next one.          */
filename auditlog;

/* Match the userid to a more Common Name to improve reporting. */
/* Updated 11/13/2013.                                          */
proc sql noprint;
  update audit_trans a
    set a_identityName = (select o.name from omrextt.userid_details_v o
      where a.a_activeuserid = o.userid),
      a_identityType='Login'
    where a.a_activeuserid in (select userid from omrextt.userid_details_v
      and a.a_recordevent='New Client Connection' and a.a_identityname='');
quit;
```

You can enhance the login reports by using A_IdentityName for the grouping to produce user name information instead of a shortened user-ID variable.

ENHANCEMENTS FOR AUDIT, PERFORMANCE AND MEASUREMENT PACKAGE

In a SAS Visual Analytics environment, the APM package addresses the activity that is logged through the metadata server and the BI server ARM logs. The APM package does not capture the SAS Visual Analytics application activity that is captured in LASR Analytic Server log files or SAS Web Application Server middle-tier log files. You must enable access to these additional logging locations and process the resulting log files to capture usage information from the SAS Visual Analytics environment.

MIDDLE-TIER LOGGING FOR SAS VISUAL ANALYTICS APPLICATIONS

The default logging for SAS Visual Analytics applications captures exceptions. The logging definition for each of the applications is defined in the XML files in **SASConfig/Lev1/Web/Common/LogConfig**. More information about these logging configuration files is documented in Chapter 7, “Fine Tuning” in *SAS Visual Analytics 6.2: Administration Guide*.

The SASVisualAnalyticsTransport-log4j.xml configuration file that is installed with SAS Visual Analytics 6.2 has an example defined of creating a rolling file appender with a name of SASVisualAnalyticsTransport6.2_LASRActions.log. If this LASR_FILE appender is enabled by changing the priority value to DEBUG, the additional LASRActions log file will be created and contain information of the user name that is accessing the LASR table. These additional log files can be created for the four main applications—Transport, Designer, Explorer, and Viewer—to determine who accessed a LASR table and the date and time at which it was accessed.

You can follow this example of adding a section to SASVisualAnalyticsExplorer-log4j.xml configuration file that creates a rolling file appender for the SAS Visual Analytics Explorer usage.

```
<appender class="org.apache.log4j.DailyRollingFileAppender"
name="SAS_LASR_COMMANDS_FILE">
  <param name="encoding" value="UTF-8" />
  <param name="datePattern" value="'. 'yyyy-MM-dd" />
  <param name="append" value="true"/>
  <param name="File"
value="/opt/mis/sas94/config/Lev1/Web/Logs/SASServer1_1/SASVisualAnalyticsExplorer6
.2_LASRActions.log" />
  <layout class="com.sas.svcs.logging.CustomPatternLayout">
    <param name="ConversionPattern"
value="%d SASVisualAnalyticsExplorer [%u] %m%n"/>
  </layout>
</appender>

<!-- modify the usage of the appender to DEBUG -->
<category additivity="false" name="com.sas.lasr.command">
  <level value="DEBUG"/>
  <appender-ref ref="SAS_LASR_COMMANDS_FILE"/>
</category>
```

This custom appender emits LASR table usage records using the defined layout stanza that you can easily parse with a SAS program. Output 4 is an example of the LASRActions log file contents.

```
2013-12-03 12:08:01,614 SASVisualAnalyticsExplorer [testsas] action=ServerInfo
2013-12-03 12:08:01,855 SASVisualAnalyticsExplorer [testsas] action=ColumnInfo
name=USER.HEART
2013-12-03 12:08:10,372 SASVisualAnalyticsExplorer [testsas] action=ServerInfo
2013-12-03 12:08:10,635 SASVisualAnalyticsExplorer [testsas] action=ColumnInfo
name=USER.AUDIT_TRANSACTIONS
2013-12-03 12:08:12,516 SASVisualAnalyticsExplorer [testsas] action=NumRows
name=USER.AUDIT_TRANSACTIONS
2013-12-03 12:08:12,622 SASVisualAnalyticsExplorer [testsas] action=NumRows
name=USER.AUDIT_TRANSACTIONS
2013-12-03 12:08:12,774 SASVisualAnalyticsExplorer [testsas] action=TableInfo
name=USER.AUDIT_TRANSACTIONS
2013-12-03 12:08:14,074 SASVisualAnalyticsExplorer [testsas] action=Summary
name=USER.MISBITT.AUDIT_TRANSACTIONS "TempComputedColumnExpression=_vae_startdt_1d5
= floor('startdt'n);
_vae_A_DateTime_1d5 = floor('A_DateTime'n);
```

```
" "TempComputedColumnNames=_vae_startdt_1d5,_vae_A_DateTime_1d5"
"VarNames=_vae_startdt_1d5,startdt,_vae_A_DateTime_1d5,A_DateTime"
2013-12-03 12:08:15,274 SASVisualAnalyticsExplorer [testsas] action=NumRows
name=USER.MISBITT.AUDIT_TRANSACTIONS
```

Output 4. LASR Table Usage for SAS Visual Analytics Explorer

The data records contain these fields: date and time of an action; application that is used; user name; and the LASR action that was executed. These actions are requests that are sent to the LASR Analytic Server. Here are some of the actions that you can monitor that show the table usage activity:

- ColumnInfo determines columns in a LASR table.
- DistinctCount determines distinct elements of one or more variables in LASR table.
- FetchRows returns rows of data from a LASR table.
- NumRows determines number of rows in a table.
- Summary derives descriptive summary statistics for one or more variables.

Whenever you access a LASR table in a report or exploration, several of the preceding actions are executed as a result of initiating report objects. In order to gauge an action such as opening a SAS data set or a LASR table, search for the Summary action to determine whether the action occurred.

Modifying APM to Process LASRACTION Log Files

After you modify the XML files, the SAS Visual Analytics applications populate additional LASRACTIONS log files. Parse the log files for information about LASR table usage. You can modify an existing APM 9.4 environment, or build separate scripts to process the log files. To add this functionality to an existing APM 9.4 installation, create a staging location for the log files in the **apm94** directory. Output 5 is an example of creating a staging location.

```
[node00] /opt/mis/sas/apm94> mkdir -p LASRACTIONS/Logs
```

Output 5. Create a LASRACTIONS Staging Directory

Log files from the various servers are moved to the **apm94** directory with the **apm94/SASEnvironment/SASCode/Jobs/runLogCopy.sas** program. There are four sections in the program to modify to enable the copy of the new log files. Near the bottom of the program, modify the “datalines;” section to contain a new entry for the lasract type.

```
login MetadataServer
wrs SASWebReportStudio
lasract LASRACTIONS
;
```

The other three sections of modifications define where the new lasract type of log files are located, how to find the datetime value in the filename, and the wildcard to use in pattern searches.

```
/* Add the new type= for the location of the LASRACTIONS log files */
%else %if &type=lasract %then %do;
    %let dir=&webDir.&slash.Logs&slash.SASServer1_1;
    %let target=&sasUsageDir.&slash.&tdir.&slash.Logs;
%end;

/* New type where date_char is derived */
else if type in ("lasract") then
    date_char=scan(copyfile,-1,".");

/* Set the wildcard pattern on LASRACTIONS log names */
else if type="lasract" then
    pattern="/_LASRACTIONS.log."/;
```

After you make these additions to runLogCopy.sas, when you execute runDataArchive.sh, the script moves the LASRActions log files from the SAS Web Application Server log location to the **apm94/LASRActions/Logs** directory. The processing is performed by adding macros to the **apm94/SASEnvironment/SASCode/Jobs/runArtifactAnalyzer.sas** program to parse the new log files.

```
%LogAuditAccess();

%LogAuditAccessSave(&loglib.,&outlib.,loadDT=&_loadtm.);

/*
 * NEW Addition ... LASRActions from Middle-Tier
 */
%put TRACE: Process LASRActionsLogs;
%let numlogs=0;
%LogUtilCleanLogLib(&loglib.);
%LogLASRActions();
```

The loglasractions.sas macro is created in **apm94/SASEnvironment/SASMacro** and is patterned after the logauditprocess.sas program. It uses existing APM macros to determine the list of log files to process and requires a new macro to parse the format of the LASRActions log files. The following macro, loglasractionsparse, is an example of log parsing code:

```
%macro loglasractionsparse(logfile=,loglib=,lasrwork=);

data &lasrwork. (keep=UserName A_DateTime A_Date Client_Context LASRTable Action
Host id pid port LASRServer StatusMsg RunTime);
length Client_Context $200;
length UserName $60;
length A_DateTime 8;
length A_Date 8;
length A_Time 8;
length Rest_Of_Line $3200;
length Action $32;
length host $32;
length ActionPart LASRTable $100;
format A_Date yymmdd10.;
format A_Time time12.;
format a_datetime datetime23.2;
format tenchars $10.;
format twelvechars $12.;
/* The following variables are used when combining the LASRActions data with */
/* LASR Server log file detail information. */
length id pid port runtime 8;
length LASRServer $200;
length StatusMsg $200;
retain linecount 0;
/* Format of a line from all 4 data sources is:
*/
/* yyyy-mm-dd hh:mm:ss,sss VisualAnalyticsAppName [firstname lastname] action=xxxx
name=lasrtablename */
/* Only action= to process are NumRows, DistinctCount, Summary and only read first
line of it */
infile "&logfile" misover firstobs=1 dsd lrecl=3200 recfm=v length=reclen;
input TenChars $10. @12 TwelveChars $12. @;
varlen=reclen - 24;
input @25 Rest_of_Line $varying3200. varlen;

/* Check first 10 chars and if not a valid date, then delete the line */
if substr(tenchars,5,1) ^= '-' and substr(tenchars,8,1) ^= '-' then delete;

A_date=input(tenchars,yymmdd10.);
TwelveChars=translate(twelvechars,'.',' ');
```

```

A_Time=input(twelvechars,time12.);
a_char=put(a_date,date9.)||":"||put(a_time,time8.);
A_Datetime=input(a_char,datetime20.);
Client_Context=scan(rest_of_line,1);
/* Now, should have only log lines that do not wrap and that have an action and
table name */
/* Find the [] for the username */
firstpos=index(rest_of_line,'[')+1;
secondpos=index(rest_of_line,']');
UserName=substr(rest_of_line,firstpos,secondpos-firstpos);
if index(UserName,'action=') > 0 then delete;
actpos=index(rest_of_line,'action=');
namepos=index(rest_of_line,'name=');
if (actpos=0) or (namepos=0) then delete;
else do;
    action=upcase(substr(rest_of_line,actpos+7,(namepos-actpos)-7));
    rest_of_line=substr(rest_of_line,namepos+5);
    lasrtable=scan(rest_of_line,1," ");
    host="&syshostname.";
    if index(action,'SUMMARY') > 0 or
        action in ('NUMROWS','DISTINCTCOUNT','HISTOGRAM','BOXPLOT','SCATTERPLOT')
then
    output;
end;
run;

%mend; /* loglasractionsparse */

```

After the log files are parsed and archived, the cumulative data set that is saved in apm94/Data/artifacts is loaded to a LASR Analytic Server library to build system usage reports.

METADATA QUERIES

In addition to the metadata information that is provided by APM, IT groups can enhance the level of detail with additional queries added to the APM infrastructure. The **apm94/SASEnvironment/SASCode/Jobs/MetadataGetArtifacts.sas** program is defined to include macro calls for artifact queries. By default, APM is installed with an invocation of **apm94/SASEnvironment/SASMacro/metadatagetartifacts**:

```

libname artifact 'Data/artifacts';
%let libref=artifact;
/*****
/* _FastMetadataGet = 1; Enable _FastMetadataGet by default to improve query */
/* response time and reduce the size of the content returned by the metadata */
/* query, specifically: eliminates selected metadata content such as: */
/* "Desc", "MetadataCreated", "MetadataUpdated" and "TargetClassifierMaps" data */
/* from the metadata content */
%let _FastMetadataGet=1;
%MetadataGetArtifacts(&libref);

```

For a SAS Visual Analytics environment, here are some metadata queries that enhance reporting:

- users, groups, and their roles
- row-level security assigned to LASR Tables
- SASIOLA and SASHDAT lists of libraries

These extra queries are created as additional macros and are invoked in the metadatagetartifacts.sas program:

```

/* IT modifications */
/* These macros were first copied to apm94/SASEnvironment/SASMacro */
%MetadataQueryLibrary(&libref);

```

```
%MetadataQueryUsers(&libref);
%MetadataQueryRowLevel(&libref);
```

You can use MetadataQueryLibrary to extract a list of all defined SASIOLA (LASR) libraries. You can use this information as input to query all the loaded LASR tables. Use the following example to query the list of libraries that are defined in metadata:

```
%macro metadataquerylibrary(lib);

data meta_libraries;
length uri serveruri conn_uri lib_uri prop_uri domainuri src_uri libname
       ServerContext AuthDomain port tag hpath opt sqldatasrc orapath host
       install_loc LASR_Name $256 desc $200 libref engine $8 isDBMS $1;
keep libname desc libref engine ServerContext AuthDomain port tag hpath opt
     sqldatasrc orapath host install_loc LASR_Name;
nobj=.;
n=1;
uri='';
serveruri='';
conn_uri='';
lib_uri='';
prop_uri='';
domainuri='';
src_uri='';

/* Determine how many libraries exist. */
nobj=metadata_getnobj("omsobj:SASLibrary?@Id contains '.'",n,uri);
/* If they exist, Retrieve the attributes for all libraries. */
if n>0 then do n=1 to nobj;
  libname='';
  ServerContext='';
  AuthDomain='';
  desc='';
  libref='';
  engine='';
  LASR_Name='';
  isDBMS='';
  port='';
  tag='';
  hpath='';
  opt='';
  sqldatasrc='';
  orapath='';
  host='';
  install_loc='';
  nobj=metadata_getnobj("omsobj:SASLibrary?@Id contains '.'",n,uri);
  rc= metadata_getattr(uri, "Name", libname);
  rc= metadata_getattr(uri, "Desc", desc);
  rc= metadata_getattr(uri, "Libref", libref);
  rc= metadata_getattr(uri, "Engine", engine);
  rc= metadata_getattr(uri, "IsDBMSLibname", isDBMS);
  rc= metadata_getprop(uri, "Library.LASR.Property.Tag.Name.xmlKey.txt", tag);
  rc= metadata_getprop(uri, "Library.SASHadoop.Path.Name.xmlKey.txt", hpath);
  rc= metadata_getprop(uri, "Library.SAS.Property.OptionString.Name.xmlKey.txt",
opt);

  /** Get associated ServerContext */
  i=1;
  rc= metadata_getnasn(uri, "DeployedComponents", i, serveruri);
  if rc > 0 then rc2= metadata_getattr(serveruri, "Name", ServerContext);
  else ServerContext='';
```



```

/* Get LibraryConnection and information within */
i=1;
rcc= metadata_getnasn(uri, "LibraryConnection", i, conn_uri);
j=1;
if engine="SASIOLA" then do;
if rcc > 0 then rc2= metadata_getattr(conn_uri, "Port", port);
else port='';
rc= metadata_getprop(conn_uri, "Connection.LASR.Property.Server.Name.xmlKey.txt",
host);
rc3 = metadata_getnasn(conn_uri, "Source", j, src_uri);
if rc3 > 0 then
    rc4=metadata_getattr(src_uri, "Name", LASR_Name);
else LASR_Name='';
end;

if engine="SASHDAT" then do;
    rc= metadata_getprop(conn_uri,
"Connection.Hadoop.NameNode.Property.Server.Name.xmlKey.txt",host);
    rc= metadata_getprop(conn_uri,
"Connection.Hadoop.Property.InstallLocation.Name.xmlKey.txt",install_loc);
end;

/* If the library is a DBMS library, get the Authentication Domain */
/* associated with the DBMS connection credentials */
if isDBMS="1" then do;
if rcc > 0 then do;
    rc2= metadata_getnasn(conn_uri, "Domain", i, domainuri);
    if rc2 > 0 then
        rc3= metadata_getattr(domainuri, "Name", AuthDomain);
        rc4= metadata_getprop(conn_uri,
"Connection.SQL.Property.DataSrc.Name.xmlKey.txt", sqldatasrc);
        rc5= metadata_getprop(conn_uri,
"Connection.Oracle.Property.PATH.Name.xmlKey.txt",orapath);
    end;
end;
output;
end;

/* If there aren't any libraries, write a message to the log */
else put 'There are no libraries defined in this metadata repository.';
run;

data &lib..libraries;
    set work.meta_libraries;
run;
%mend;

```

LOADED LASR TABLE QUERIES

With the collection of a list of SASIOLA libraries, you can assemble another query to retrieve the table attributes of all the loaded LASR tables. Execute the query by using the same user ID that started the LASR Analytic Server. Following this guideline, the LIBNAME statement to SASIOLA does not require the SIGNER= option and improves the performance of the query execution. The following code provides an example of querying the characteristics of members of a LASR library:

```

%let libref=TESTLASR;
Libname &libref. SASIOLA tag="user.testlasr" port=10010 host="node00.xxx.yyy.com";
proc sql noprint;
    create table one_list as
    select memname, memlabel, crdate, modate, nob, obslen
    from sashelp.vtable where libname="&libref.";
quit;

```

If you execute this code against all the SASIOLA libraries and you collect and archive the data daily, then this table of information is loaded into the SAS Visual Analytics environment. This data provides a look at the table size growth over time. If you join this table with the LASR table access counts that you are collecting, then you can view growth over time as well as the access count over time. This provides the capability to watch the larger LASR tables to ensure they are being actively used in the environment. Display 10 is a section of a cross tabular report to examine size and usage.

Date ▲	Nov15/2013		Nov18/2013		Nov19/2013
LASR Table ▲	Size (GB)	Access Count	Size (GB)	Access Count	Size (GB)
Total	55.62852	63820	49.19635	62114	50.534 ▲
IT.AUDIT_TRANSALL	6.58367	618	6.58367	619	6.822
IT.IT_PROJECT_ACTUALS	0.27749	38982	0.27836	38987	0.279
IT.IT_PROJECT_ALLOCATIONS	0.06934	17688	0.06942	17712	0.069
IT.LASR_TABLES	0.00025	163	0.00026	164	0.000
IT.LASR_TABLES_HISTORY	0.01765	373	.	.	0.018
IT.LASR_USAGE	1.05857	1350	.	.	1.069
IT.OVERALL_WORKLOAD
IT.POS_DATA

Display 10. LASR Table Size and Access Count over Time

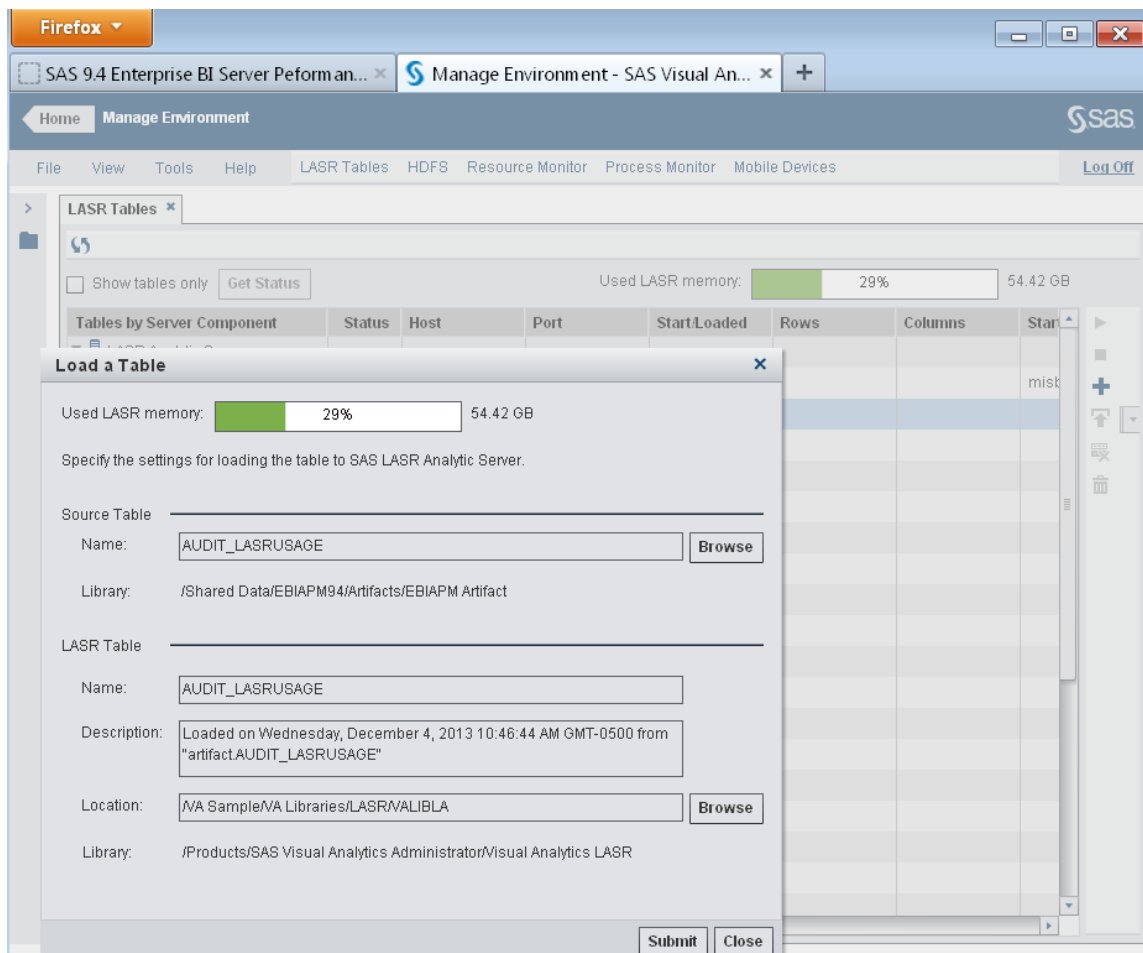
EXAMINE SYSTEM USAGE WITH SAS VISUAL ANALYTICS REPORTS

Now you have reached the fun part of building your system auditing reports in SAS Visual Analytics. You have completed the framework to capture login information, LASR table usage, and system and metadata information. These data sources provide a snapshot of system activity that can be surfaced to the system users, auditors, and IT management. Here are some system usage questions that this infrastructure can address:

- Who is accessing the system and which applications do they use the most?
- What are the most active times for system logins and table usage?
- Which tables are accessed the most and by which group?
- What are the largest tables loaded to LASR and how often are they consumed?
- Can we determine the trend for table size growth and forecast our system capacity?

LOADING TABLES INTO A LASR SERVER

Before using a data source in a SAS Visual Analytics environment, you register the data set metadata and load the data into a LASR library. Display 11 is an example of loading a new table into a LASR library using the SAS Visual Analytics Administrator application.



Display 11. Adding AUDIT_LASRUSAGE into a LASR Library

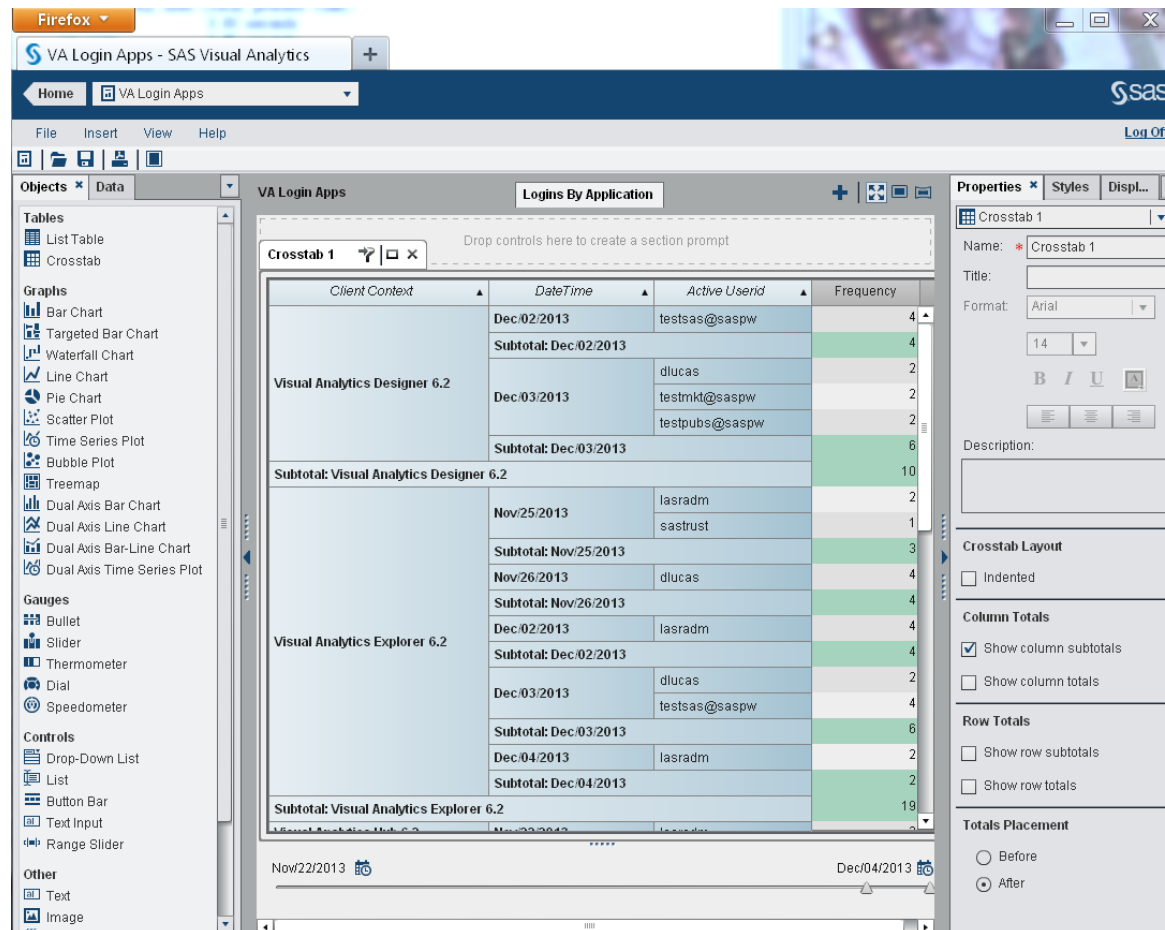
After loading the data set into a LASR library, you can explore the data from SAS Visual Analytics applications. You can see the available table that contains the LASR table usage information in Display 12.

Tables by Server Component	Status	Host	Port	Start/Loaded	Rows	Columns	Acc...
LASR Analytic Server							
Connection: LASR Analytic ...	●	misvad00.unx.s...	10010	12/02/13 04:05...			12/0...
Visual Analytics LASR							
SASHELP_CARS	●			12/03/13 05:04...	428	15	12/0...
SASHELP_HEART	●			12/03/13 05:05...	5209	17	12/0...
AUDIT_TRANSACTI...	●			12/04/13 10:25 ...	280870	26	12/0...
AUDIT_LASRUSAGE	●			12/04/13 10:51 ...	141	14	12/0...

Display 12. AUDIT_LASRUSAGE Available for Usage

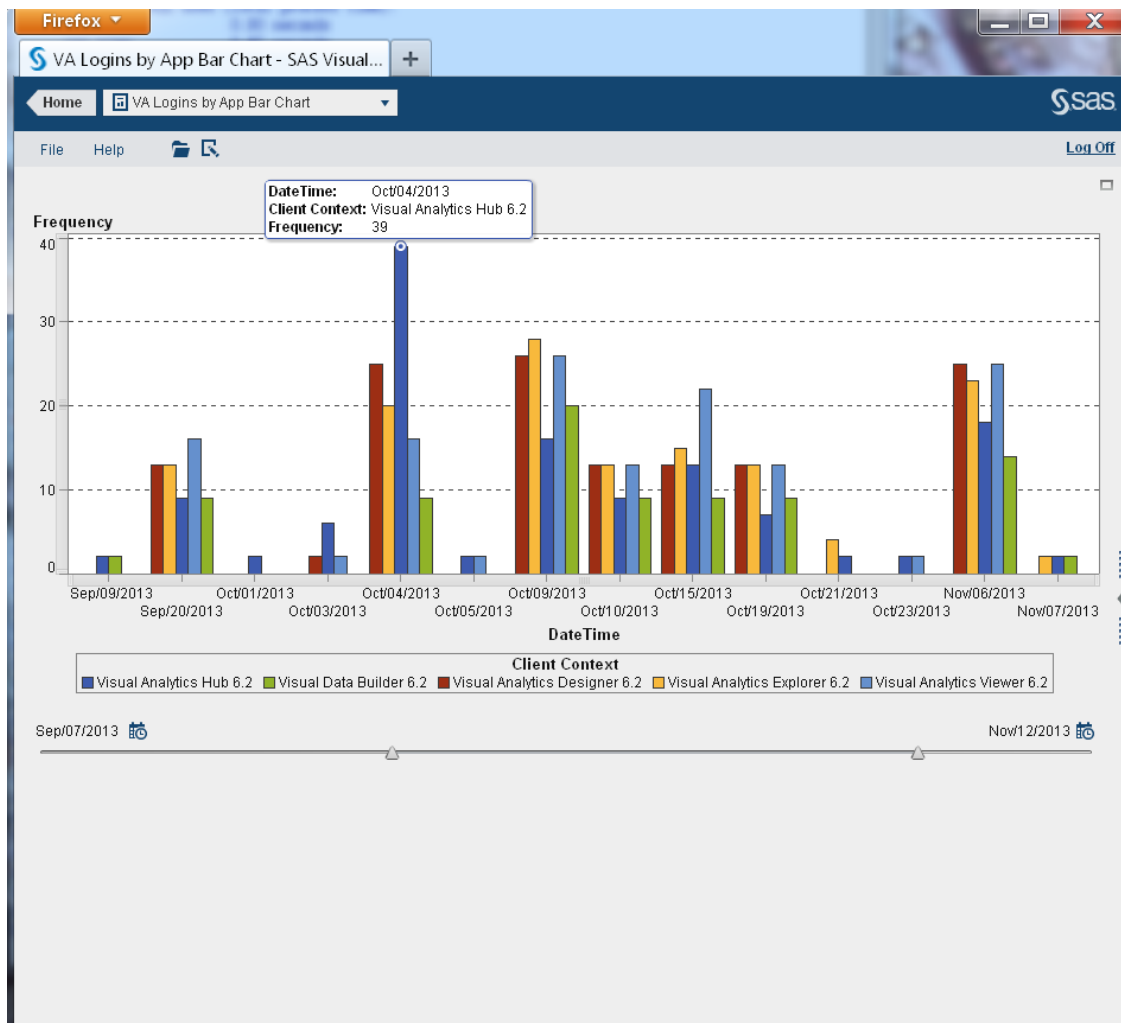
USER LOGINS

One of the first questions to address with the APM package is the user login information. With the same data set loaded into a SAS Visual Analytics environment, you can dive a little deeper into the login details. Similar to the steps from “Exploring the Login Data” in an earlier section, you can build a quick report by focusing on the Client Context, DateTime, and Active User ID variables that are part of the Audit_Transactions table. Display 13 shows a simple report of these three fields and a Range Slider control that has an interaction with the cross-tabular report.



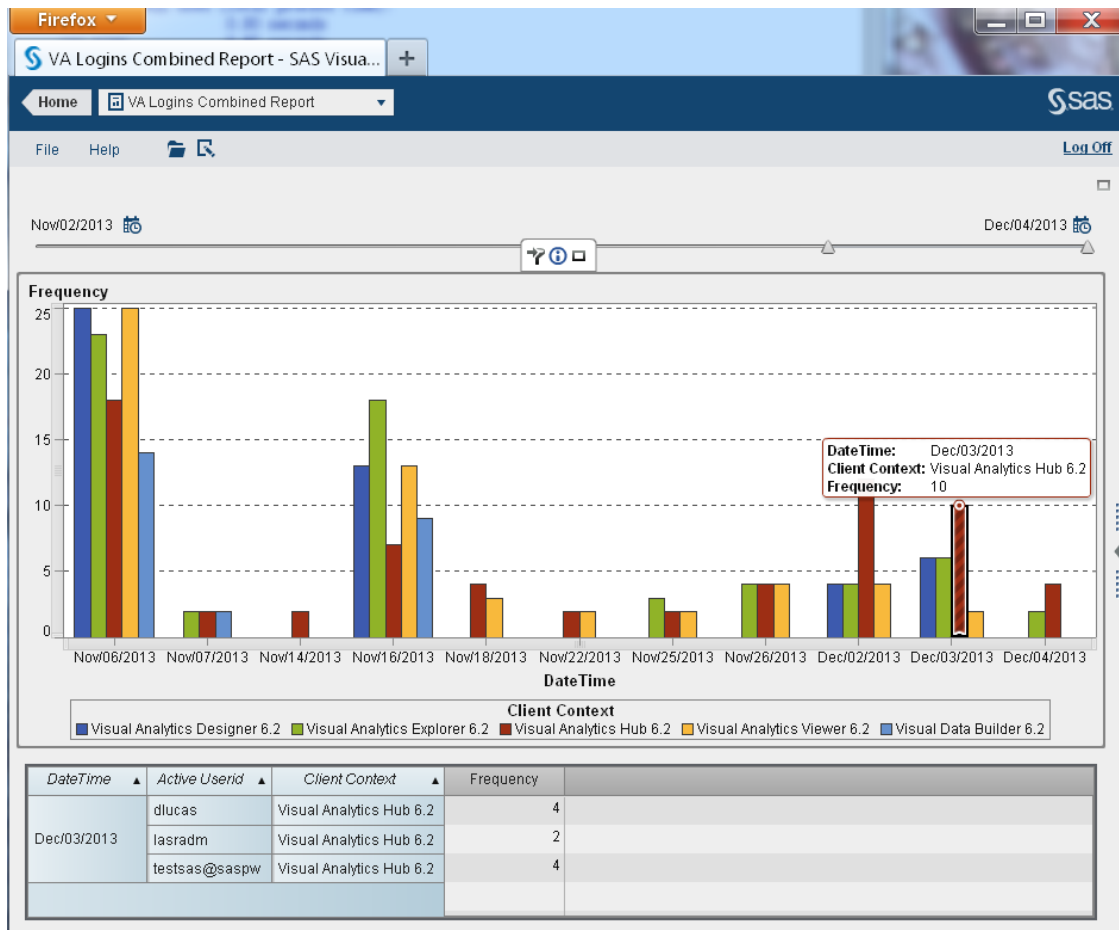
Display 13. Designer Report for Users Accessing System by Application

The range slider using the DateTime column allows the interaction between the slider and cross-tabular report, which enables you to subset by date range. Another way you can view this same type of data to gauge growth over time is to plot the logins by application with a bar chart. Display 14 is a bar chart using DateTime as a category and grouping by Client Context. If you place your mouse pointer over the bar chart, you can see the value for the data element.



Display 14. SAS Visual Analytics Report Viewer for Logins by Application over Time

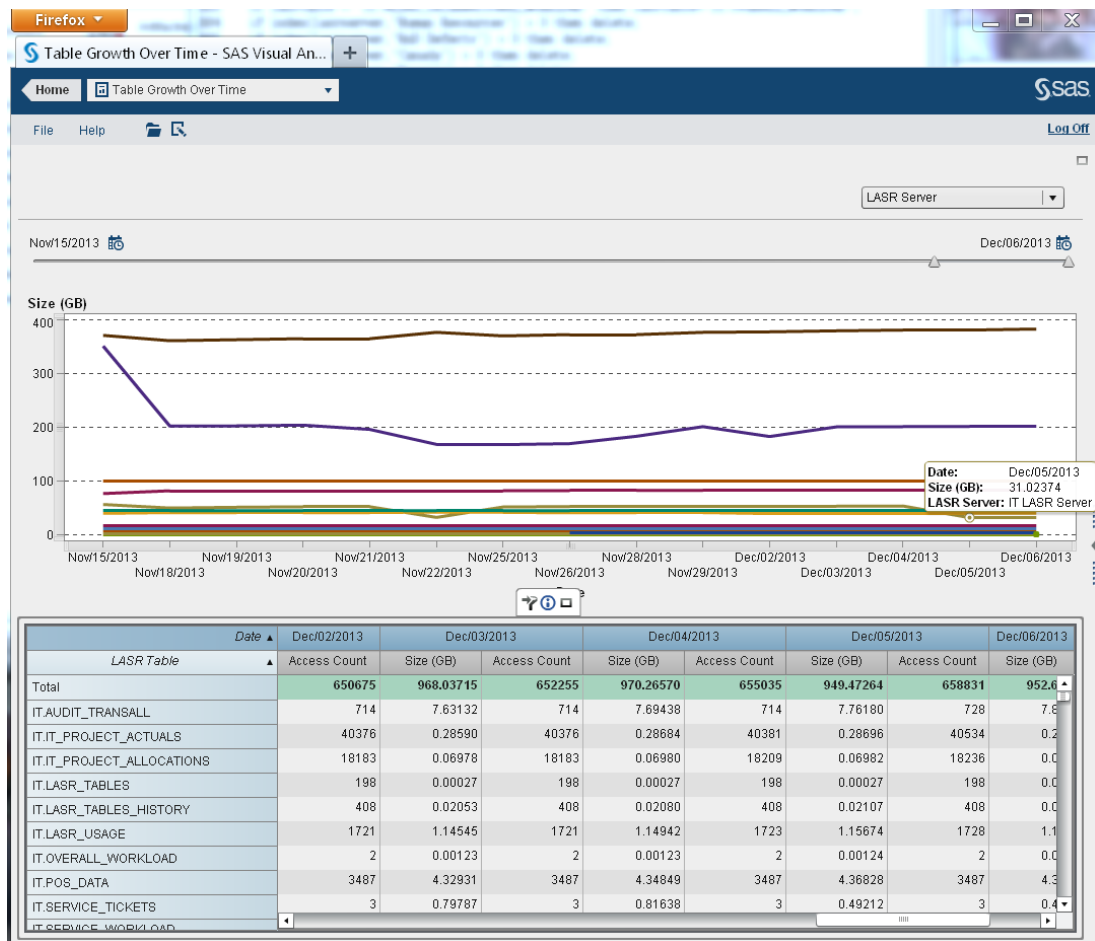
With the combination of these two reports, you can review your login amounts to determine system usage growth and drill to the detail that shows who is using the system on these dates. Display 15 contains the two report objects with interactions linking the two filters to expand the bar chart to show greater detail.



Display 15. Bar chart interaction with user login details

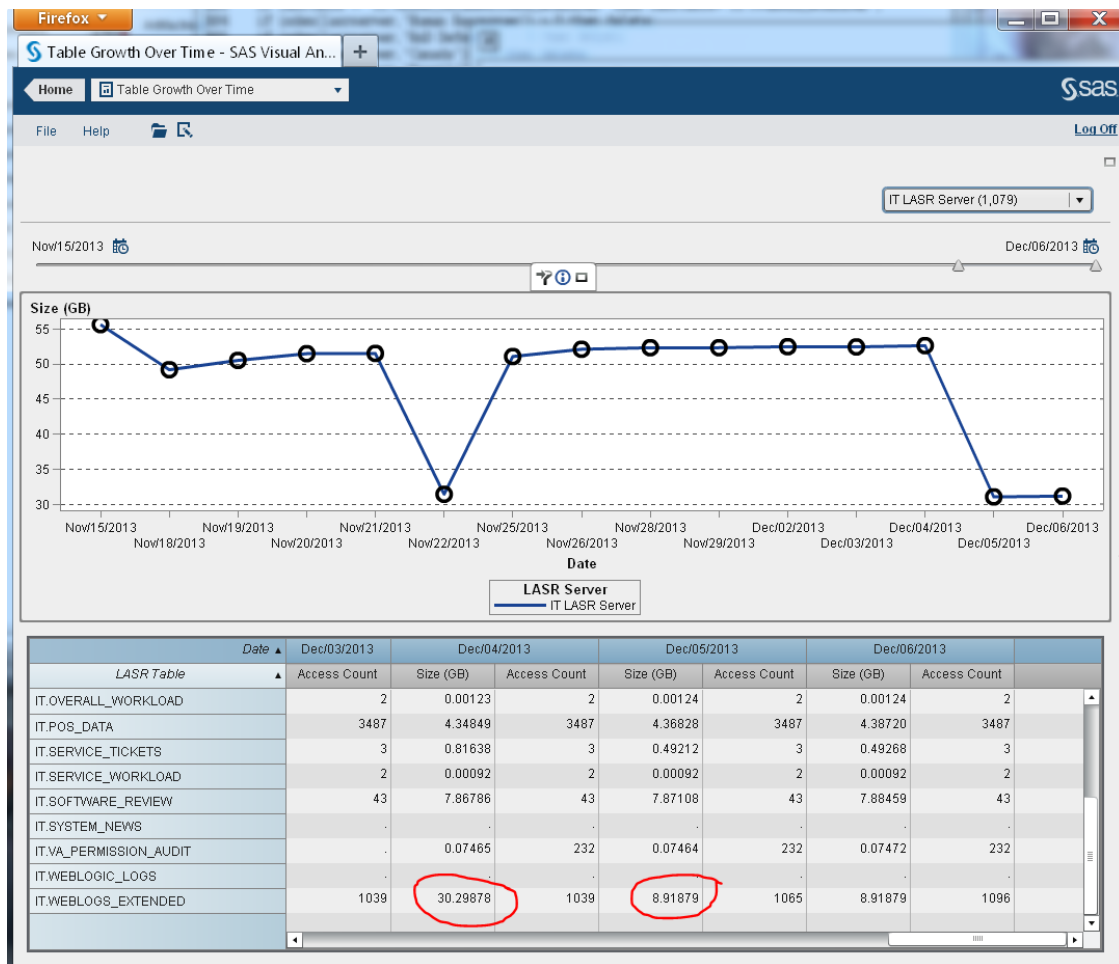
LASR TABLE USAGE

Similar concepts for LASR table consumers are used to determine the LASR tables with the highest access count. With the creation of the Audit_LASRUsage data set and contents loaded into a LASR library, SAS Visual Analytics explorations and reports can track system usage. In the example code to query the loaded LASR tables in a SASIOLA library, you can determine which tables belong to a LASR Analytic Server. This table-to-server mapping gives the ability to create a simple report that provides the main consumers of LASR tables that are associated with a LASR Analytic Server. Display 16 is an example of how to develop a report with the ability to subset by LASR Analytic Server name by using a drop down list. This example enables you to review all of the users of the LASR tables that are associated with the server.



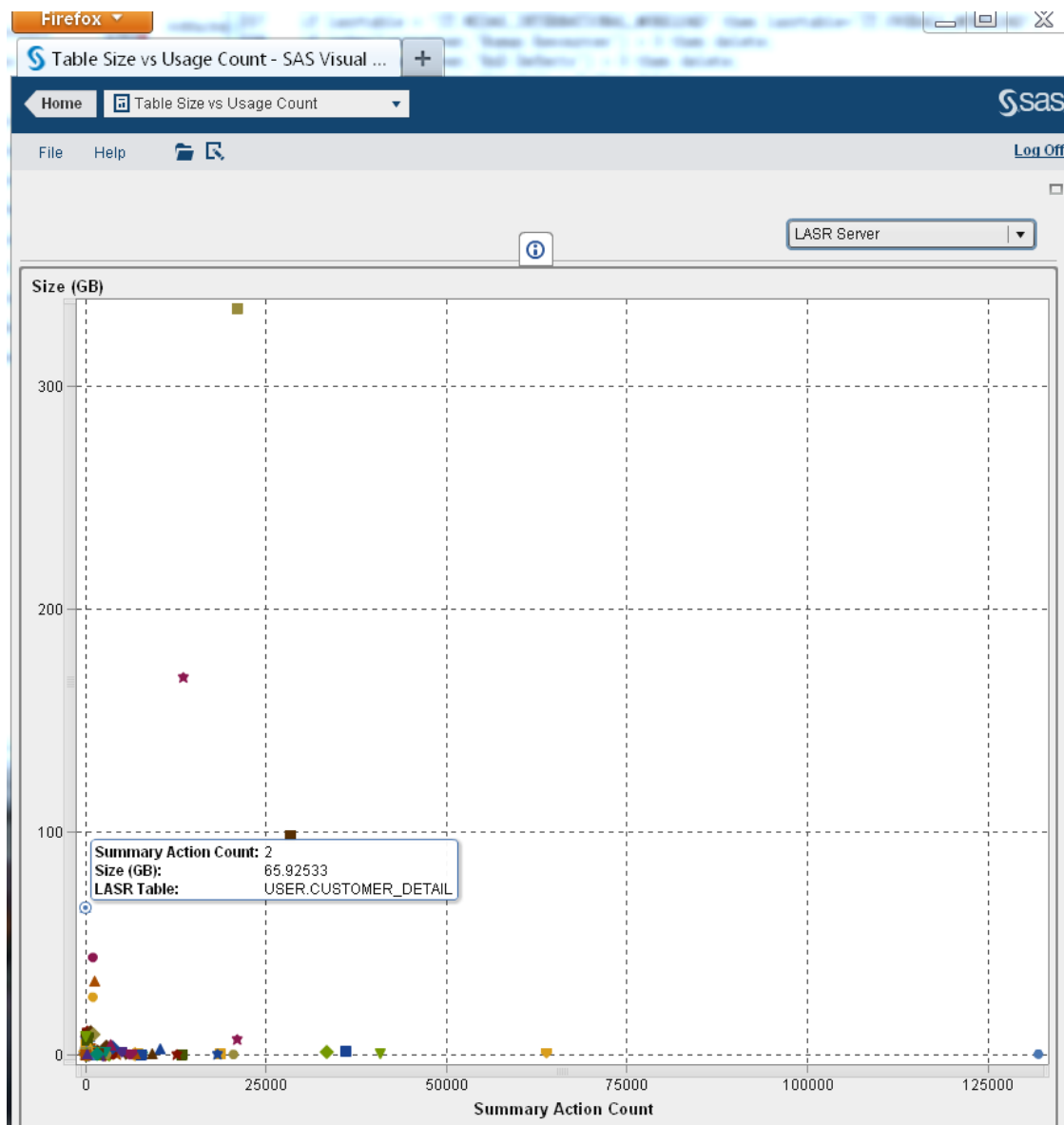
Display 17. Review LASR Table Total Memory and Growth over Time

There was steady growth of memory usage of approximately 2 gigabytes. Then on December 5, there was a decline of around 21 gigabytes. By examining the line chart with the downward slope on that day, you can determine that IT LASR Analytic Server data decreased. Dive deeper into the analysis by using the drop down filter to subset on IT LASR Analytic Server and then review the change in December 4 and December 5 table lists. Display 18 provides the detail where the LASR table IT.WEBLOGS_EXTENDED decreased in size from 30.2 gigabytes to 8.9 gigabytes.



Display 18. Determine Decrease in LASR Memory Usage

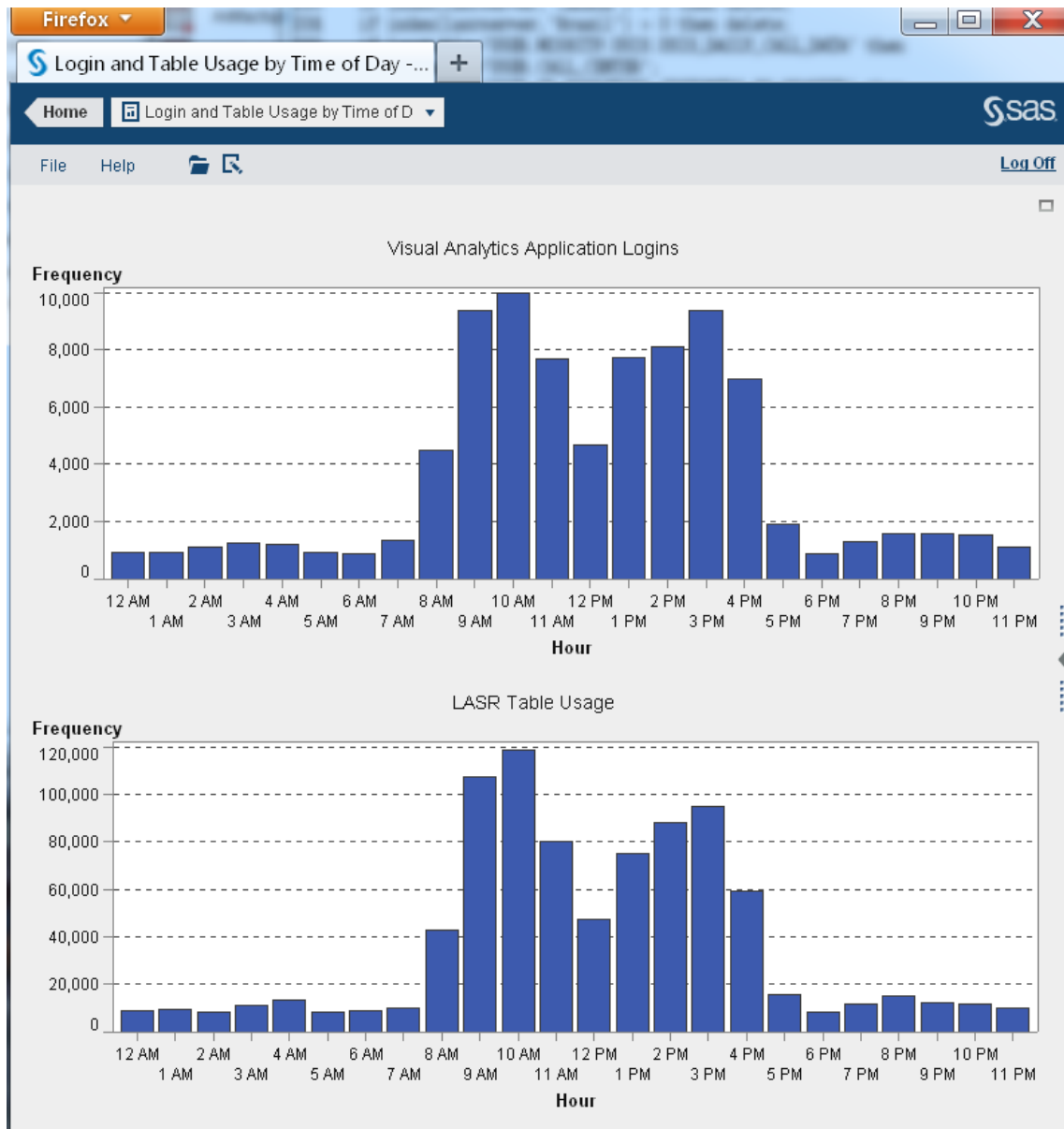
When memory usage reaches the point where additional computer nodes might be required, you can use these same tables to review the LASR table sizes and cross reference to the number of accesses of a table. A great way to visualize the information is through a simple scatter plot that uses the table size and access count, differentiating the LASR tables by color. Display 19 reveals that the User.CUSTOMER_DETAIL table, which is 65 gigabytes, has been referenced 2 times. This table is a likely candidate for removal from the system until the users of the table are ready to use the data source. This change prevents the need for additional hardware.



Display 19. Large LASR Table with Low Utilization

SYSTEM PEAK TIME

With an Enterprise SAS Visual Analytics environment, an administrator has to find low usage windows of time in which to schedule backups, data refreshes to LASR Analytic Servers, and other system tasks that might impact performance. With the collection of data for system logins and table usage, you can combine the two data sources to review a snapshot of peak system usage. After these windows of time are determined, you can select windows of opportunity for scheduling processes. For this type of report, instead of focusing on a day, the grouping is done by hour. Display 20 shows two bar charts that examine table access and logins by hour.



Display 20. System Usage by Time of Day

CONCLUSION

To support a SAS Visual Analytics environment, the IT division and user groups need to inquire about system and data usage across the enterprise. These questions probe the number of users, security, table sizes, and a myriad related questions. The administrator of the system must capture the data and present the results to the varied groups of inquirers. This paper presents methods for using the SAS APM framework to capture these various data sources. After the data is captured and loaded to LASR tables, the SAS Visual Analytics environment can be used as a report delivery and exploration tool. The discussion of some of these basic methods to query system information and browse log files provides some techniques that can be customized to fit the needs of your IT group. When you understand these concepts, the ability to build your own reports to dig deeper into the details about system usage is just a visualization away.

ACKNOWLEDGMENTS

We would like to thank Jason Shoffner for his editing contributions to this paper.

RESOURCES

SAS Institute, Inc. 2013. *SAS® LASR™ Analytic Server 2.1: Administration Guide*. Cary, NC: SAS Institute, Inc.

SAS Institute, Inc. 2013. *SAS® Visual Analytics 6.2: Administration Guide*. Cary, NC: SAS Institute, Inc.

SAS Institute Inc. 2013. *Audit, Performance and Measurement Package for SAS 9.4 Enterprise Business Intelligence*. Available at <http://support.sas.com/rnd/emi/APM94/index.html>. Accessed on November 12, 2013.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors:

Dan Lucas
SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
919-531-6551
Dan.Lucas@sas.com

Brandon Kirk
SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
919-531-3825
Brandon.Kirk@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.