Paper 10160-2016

Design for Success: An Approach to Metadata Architecture for Distributed Visual Analytics – 10160

Ratul Saha, Vimal Raj, and Vignesh Balasubramanian, KAVI Global, Barrington, IL

ABSTRACT

Metadata is an integral and critical part of any environment. Metadata facilitates in resource discovery and provides a unique identification of every single digital component of a system, simple to complex. SAS® Visual Analytics being one of the most powerful analytics visualization platform, leverages the power of metadata to provide a plethora of functionalities for all types of users. The possibilities range from real-time advanced analytics and power-user reporting to advanced deployment features for a robust and scalable distributed platform to internal and external users. This paper explains the best practices and advanced approaches to design and manage the metadata for a distributed global SAS Visual Analytics environment.

Designing and building the architecture of a distributed SAS® Visual Analytics environment requires attention to various important factors like user groups and roles, access management, data protection, data volume control, performance requirements and so on. This paper covers how to build a sustainable and scalable metadata architecture through a top-down hierarchical approach. It helps SAS® Visual Analytics Data Administrators to improve the platform benchmark through memory mapping, perform administrative data load (AUTOLOAD, Unload, Reload-on-Start and so on), monitor artifacts of SAS® LASR™ Analytics Servers on co-located Hadoop Distributed File System (HDFS), optimize high-volume access via FullCopies, build customized FLEX themes, and so on. It showcases practical approaches to managing distributed SAS LASR Analytics Servers, offering guest access for global users, managing host accounts, enabling Mobile BI, using power-user reporting features, customizing formats, enabling home page customization and using best practices for environment migration.

INTRODUCTION

Choosing a BI implementation that fits your organization’s need is very important. The right platform and enabled toolset motivates employees, provides important insight at the right time, drives faster decisions, provides value and puts your company in a forward-looking path. The path leading to a sustainable, robust and scalable infrastructure can reap benefits from evolving technologies over time. SAS Visual Analytics provides a plethora of options to design that suitable BI platform that works for you. Its in-memory analytics, modeling, and reporting capabilities provide you the right information when you need it, how you need it.

A standard mid-to-large size organization which drive decisions based on data analyzed in SAS Visual Analytics, usually have dedicated reporting/analytic unit that is responsible for analyzing information from various diversified data systems for dynamic reporting accessible to multiple secure devices. To support the fast in-memory data and analytical processes, we need to think beyond the traditional approaches and deploy optimally performing hardware and software techniques for exemplary results. Designing the metadata and security is an integral part of building this robust Visual Analytics platform. This paper will empower the SAS Administrators with metadata design approaches and best practices to streamline their Visual Analytics reporting process. We expect our readers to have intermediate knowledge about metadata administration in SAS and Visual Analytics 7.2 environment.

METADATA ARCHITECTURE OF A VISUAL ANALYTICS ENVIRONMENT

In designing the detailed metadata architecture for Distributed Visual Analytics, we have explained the below subject areas. Details of each section are explained below.

1. User Roles
2. User Groups and Access
3. Process Streamlining
4. Distributed VA server architecture
5. LASR server architecture
6. Metadata folder structure
7. Metadata and content security
8. Best Practices for Metadata Architecture Design

USER ROLES

This section explains the different set of users based on their performing roles in the Visual Analytics environment. These roles will largely define the groups the users belong to and their role-specific access requirements. This is the stepping stone of designing right form of access around the distributed and shared VA metadata across an internal and external group of users.

Role	Responsibilities	Performing Area	Performer(s)
System Administration	- Infrastructure management - Server management - Resource monitoring - Maintenance - User Management and LDAP Access - Metadata backup - OS level file, folder and access management - Software patch install and software update	Throughout all environments	Organization Admins
Metadata and Data Administration	- Future metadata infrastructure design - Managing libraries (SAS base, Oracle etc.) - Datamart metadata updates (name change, security update etc.) - Metadata folder management - Metadata object promotion - Data promotion - Hadoop data management - Manage themes	Throughout all environments' metadata including Hadoop	Functional Admins and Implementers
VA Content Creation	- Integrating Datamart and local data into VA memory - VA data preparation - Create/Publish reports and explorations	To specific locations at VA metadata (development environment only)	Functional Developers
VA Content Consumption	- Consuming VA reports and explorations from specific preauthorized locations	To specific locations at VA metadata (production environment only)	Functional and Global Users
Testing	- Testing changes in metadata, data, and contents during enhancement, promotion and bug fix. - Documenting test results for further operation	Throughout Test environment only	Functional Testers

Table 1. Defining user roles

USER GROUPS AND ACCESS

This section explains metadata user groups and access types for each set of users. Note the accesses are different across environments to best suit their needs. We assumed that there would be separate content creators and consumers based on organization business function with logical separation.

Group Name	Members	Role	Access Type
System Admin	Organization System Administrators Only	System Administration	RWX – Open across all environments
Metadata and Data Admin *	Data Architects from organization who are expert on data subject areas and have knowledge about individual project data requirements	Metadata and Data Administration	RWX – Limited (lower level access than System Admin) across all environments
Developer – Group 1	Programmers, analysts, modelers, statisticians and ETL developer group for Business Function 1	VA Content Creation	RW – Restricted to only certain area of metadata into

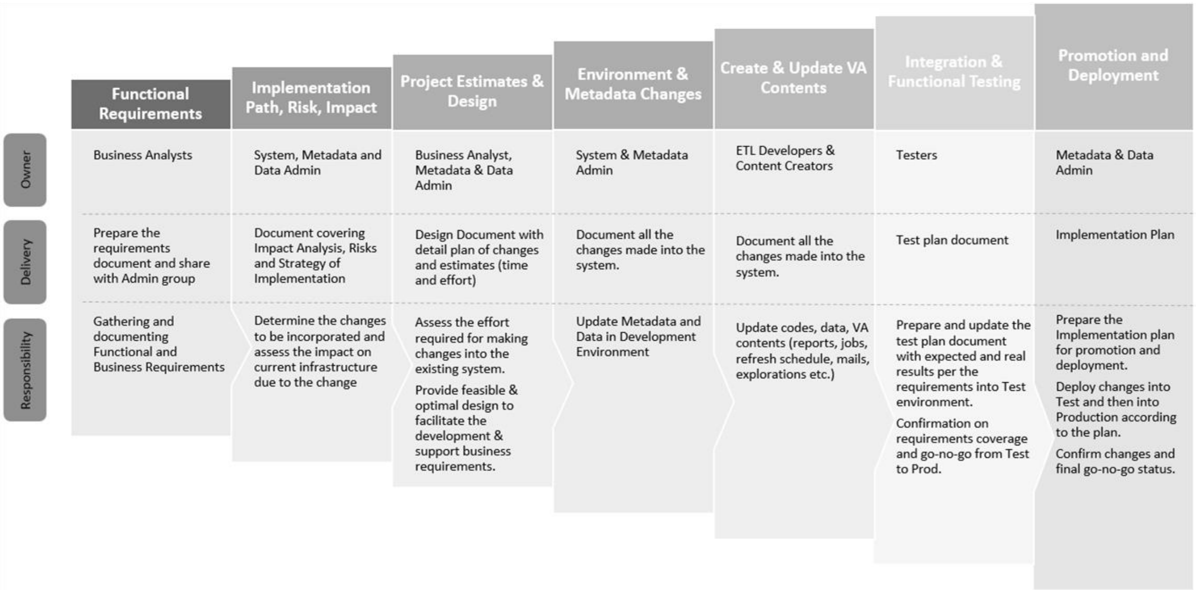
Developer – Group 2	Programmers, analysts, modelers, statisticians and ETL developer group for Business Function 2		Development environment
Developer – Group 3	Programmers, analysts, modelers, statisticians and ETL developer group for Business Function 3		
User – Business Function 1	Content Consumers for business function 1	VA Content Consumption	R – Restricted to designated metadata folders only into Production environment
User – Business Function 2	Content Consumers for business function 2		
User – Business Function 3	Content Consumers for business function 3		
Global Users	No specific login is required as contents can be access		
Testers *	Application, data, model and metadata testers across all the business area	Testing	R - Restricted to designated metadata folders only into Test environment

Table 2. Defining user groups across roles

* Can be further sub-divided into multiple groups based on organization needs.

PROCESS STREAMLINING

It is very important to streamline the process around content management across the environment. The data governance team and security architects define the rules on the boundary of activities performed. Below is a sample process flow with process ownership, deliveries, and responsibilities.



	Functional Requirements	Implementation Path, Risk, Impact	Project Estimates & Design	Environment & Metadata Changes	Create & Update VA Contents	Integration & Functional Testing	Promotion and Deployment
Owner	Business Analysts	System, Metadata and Data Admin	Business Analyst, Metadata & Data Admin	System & Metadata Admin	ETL Developers & Content Creators	Testers	Metadata & Data Admin
Delivery	Prepare the requirements document and share with Admin group	Document covering Impact Analysis, Risks and Strategy of Implementation	Design Document with detail plan of changes and estimates (time and effort)	Document all the changes made into the system.	Document all the changes made into the system.	Test plan document	Implementation Plan
Responsibility	Gathering and documenting Functional and Business Requirements	Determine the changes to be incorporated and assess the impact on current infrastructure due to the change	Assess the effort required for making changes into the existing system. Provide feasible & optimal design to facilitate the development & support business requirements.	Update Metadata and Data in Development Environment	Update codes, data, VA contents (reports, jobs, refresh schedule, mails, explorations etc.)	Prepare and update the test plan document with expected and real results per the requirements into Test environment. Confirmation on requirements coverage and go-no-go from Test to Prod.	Prepare the Implementation plan for promotion and deployment. Deploy changes into Test and then into Production according to the plan. Confirm changes and final go-no-go status.

Figure 1. A sample streamlined process around VA project environment

Once the development and implementation process is layered out, controlling the development process around the system becomes manageable too. Below diagram shows an ideal VA content creation process. It shows the step-by-step approach for creating some of the Visual Analytics contents at their designated metadata locations.

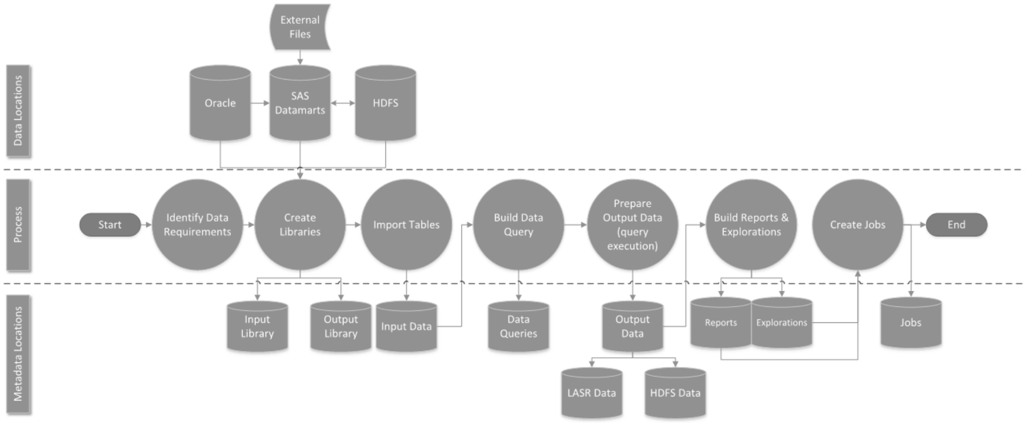


Figure 2. An ideal VA content creation process

In a large organization, where multiple project teams are working simultaneously, it's important to define designated metadata placeholders for each type of objects for easy management.

DISTRIBUTED VA SERVER ARCHITECTURE

Below diagram shows the architecture of a sample distributed VA environment.

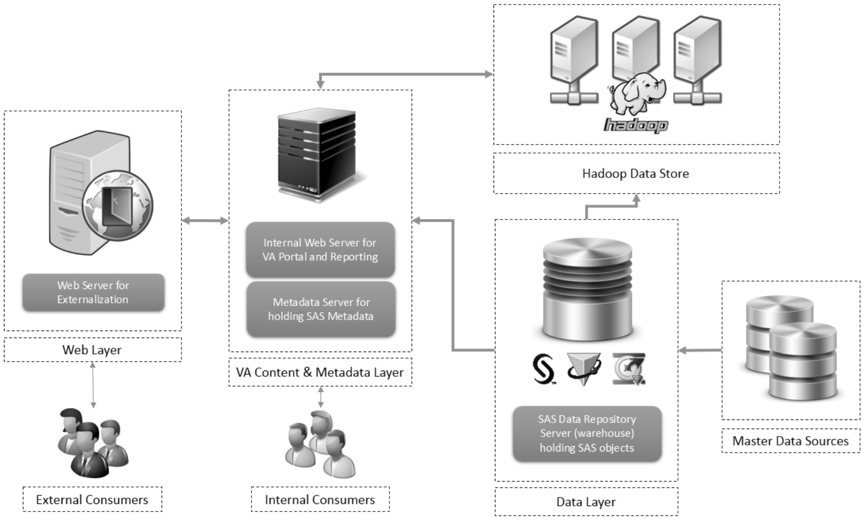

hadoop
Hadoop Data Store
Web Server for Externalization
Web Layer
Internal Web Server for VA Portal and Reporting
Metadata Server for holding SAS Metadata
VA Content & Metadata Layer
SAS Data Repository Server (warehouse) holding SAS objects
Data Layer
Master Data Sources
External Consumers
Internal Consumers

Figure 3. Distributed VA Server Architecture

In the above diagram, distributed Visual Analytics environment uses Hadoop as its in-memory data store. Hub's tools connect the Data Layer (as DWH) to Hadoop for data exchange. Reporting for internal customer happens through Hub. External (outside network) customers connect to Hub through Guest access over an external facing secure web server.

LASR SERVER ARCHITECTURE

For distributed multi-function environment, it is recommended to have dedicated LASR Analytics Servers. Individual LASR servers provide separate operational spaces for each subject area, lesser memory dependency, better performance, and ease of advanced debugging, control and management.

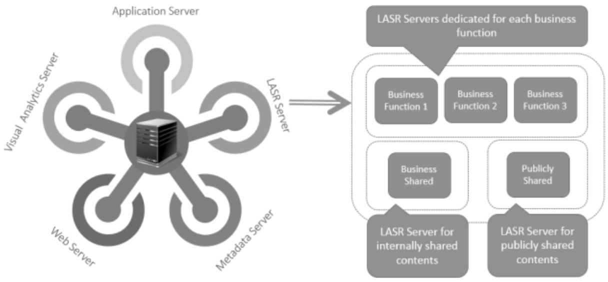


Figure 4. LASR Server Architecture

The individual LASR servers dedicated connections will have separate metadata objects linked to them from each separate business functions.

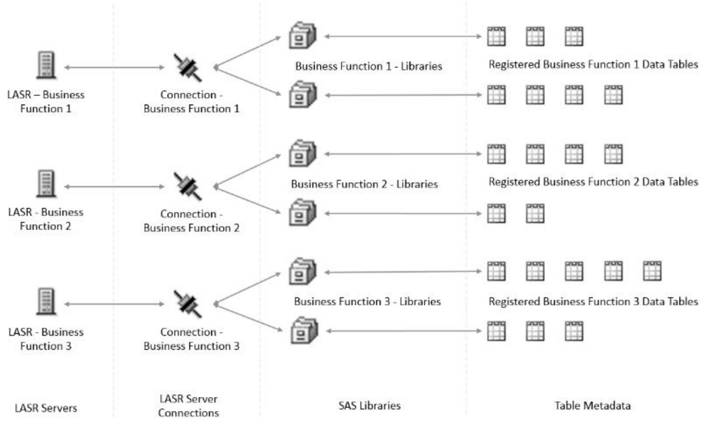

LASR – Business Function 1
Connection - Business Function 1
Business Function 1 - Libraries
Registered Business Function 1 Data Tables
LASR - Business Function 2
Connection - Business Function 2
Business Function 2 - Libraries
Registered Business Function 2 Data Tables
LASR - Business Function 3
Connection - Business Function 3
Business Function 3 - Libraries
Registered Business Function 3 Data Tables
LASR Servers
LASR Server Connections
SAS Libraries
Table Metadata

Figure 5. Connection between dedicated LASR Servers and metadata objects

Data libraries from similar subject areas will connect to a dedicated LASR server for that subject area with its own connection. This way the loads would be shared and separate.

METADATA FOLDER STRUCTURE

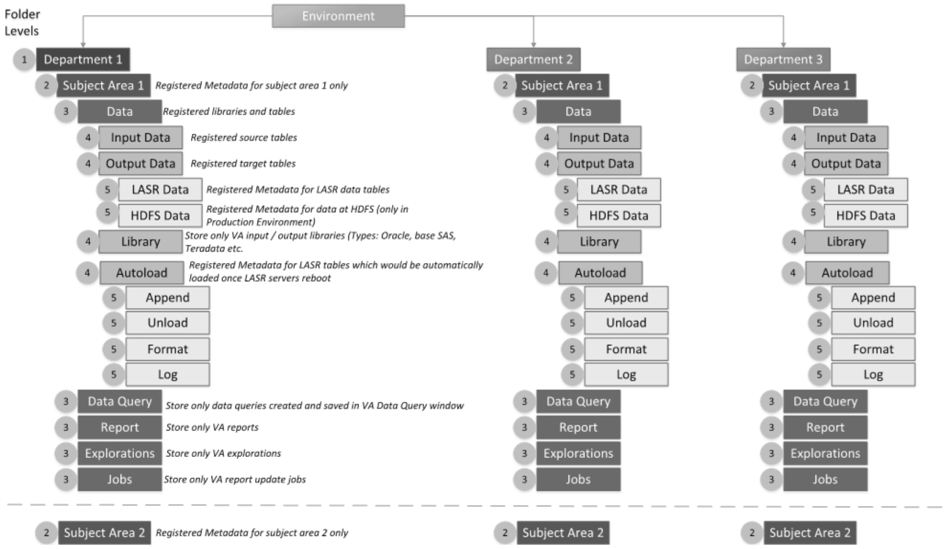

Folder Levels
Environment
1 Department 1
2 Subject Area 1
Registered Metadata for subject area 1 only
3 Data
Registered libraries and tables
4 Input Data
Registered source tables
4 Output Data
Registered target tables
5 LASR Data
Registered Metadata for LASR data tables
5 HDFS Data
Registered Metadata for data at HDFS (only in Production Environment)
4 Library
Store only VA input / output libraries (Types: Oracle, base SAS, Teradata etc.
4 Autoload
Registered Metadata for LASR tables which would be automatically loaded once LASR servers reboot
5 Append
5 Unload
5 Format
5 Log
3 Data Query
Store only data queries created and saved in VA Data Query window
3 Report
Store only VA reports
3 Explorations
Store only VA explorations
3 Jobs
Store only VA report update jobs
2 Subject Area 2
Registered Metadata for subject area 2 only
Department 2
Department 3

Figure 6. A sample metadata folder structure

The above diagram shows a sample metadata folder structure for Visual Analytics metadata contents. This would be applicable for most organizations with multiple business functions (departments). Subject area folders are placed under department folders to cluster similar subject areas under one hierarchy. Some advantages of this design are explained below.

- Fewer number of folders to manage
- Easy and faster to navigate during developing and viewing
- Easy to manage for security and administration with permission inheritance
- Data from same subject areas are kept close for easy development
- Proper segmentation of metadata objects based on object type for easy access
- Easy to replicate, less congestion
- Easy to follow during migration and promotion from one environment to other

METADATA AND CONTENT SECURITY

Once the roles are setup, the process is streamlined and metadata architecture is complete, most important step before development starts, is finalizing the security. If reports contain important data which needs to be protected from viewing or accidental overwrite, metadata folder level security is a must. This will ensure only registered and permitted groups of users are allowed to view or edit the contents of those folders and perform only permitted actions over those metadata objects.

We have created few common ACTs with specific allow and deny operations. Then we applied one or more of those ACTs to each folder to secure them. To grant access back to a particular group, we have supplemented the ACT settings by adding explicit controls on the target folder. We have applied the security ACTs by a top-down approach, which helped child-level folders to inherit securities from their parent folders.

- ACTs to LIMIT ACCESS
  - Create some general-use ACTs to reduce access to minimum
  - Apply one or more to each folder (starting from ultimate parent folder) that need to be secured
- Controls to GRANT ACCESS
  - Add Explicit Controls on Target folders to particular group(s)

Before discussing the ACTs further, let's have a look at the below table, that describes the specific operations performed by groups across each environment.

Environment	Group Category	Operation			
		Create	Read	Update	Delete
Development	Administrators	System Parameters, Metadata, Data	System Parameters, Metadata, Data	System Parameters, Metadata, Data	Metadata, Data
	Developers	All VA contents	All VA contents	All VA contents	All VA contents
Test or QA	Administrators	System Parameters, Metadata, Data	System Parameters, Metadata, Data	System Parameters, Metadata, Data	System Parameters, Metadata, Data
	Testers	Data	Metadata, Data, System, all VA contents	Data	None
Production	Administrators	System Parameters, Metadata, Data	System Parameters, Metadata, Data	System Parameters, Metadata, Data	System Parameters, Metadata, Data
	Users	None	Selected VA contents	Selected VA contents	None

Table 3. Operations performed by user groups

Each operation is a combination of some Grant permissions and some Deny permissions.

Operation	Grant	Deny
Create	Read, Create, WriteMetadata	All others
Read	Read, ReadMetadata	All others
Update	Read, Write, CheckInMetadata, Administer	All others
Delete	Read, Delete	All others

Table 4. Grant and Deny combination for each operation

Different permissions are explained below.

Permission	Actions Affected and Limitations on Enforcement
ReadMetadata (RM)	View an object or navigate past a folder. For example, to see an information map one needs RM for that information map. To see or traverse a folder one needs RM for that folder.
WriteMetadata (WM)	Edit, delete, change permissions for, or rename an object. For example, to edit a report one needs WM for the report. To delete a report one needs WM for the report (and WMM for the report's parent folder).
WriteMemberMetadata (WMM)	Add an object to a folder or delete an object from a folder. To enable someone to interact with a folder's contents but with not the folder itself, grant WMM and deny WM.
CheckInMetadata (CM)	Check in and check out items in a change-managed area. Applicable only in an optional configuration for SAS Data Integration Studio.
Read (R)	Read data. For example, while one needs RM for a cube in order to see that cube, one needs R for the cube in order to run a query against it.
Write (W)	Update data. For example, on a table, W controls updating the rows in the table.
Create (C)	Add data. For example, on a table, C controls adding rows to the table.
Delete (D)	Delete data. For example, D on a library controls deletion of tables from the library.
Administer (A)	Monitor, stop, pause, resume or refresh a server or spawner. It's an operational role, not by any permission.

Table 5. Various permissions and their details

From the maintenance point of view, managing controls and accesses at Group level are much easier than working at the user level. In that way, you just need to add users to a group to provide similar access to all users from that group. A number of Groups and ACTs needed with their names should be chosen carefully based on your need and that can be optimally managed.

In this paper, we have used three general-use ACTs. Each ACT reduces a particular type of access down to a minimal level. These ACTs can be used together to protect folders.

- HIDE: Providing only SAS administrator group and SAS trusted user exclusive READ access.

Group	Grant	Deny
PUBLIC		ReadMetadata
SAS Administrators	ReadMetadata	
SAS Systems Services (includes SAS Trusted User only)	ReadMetadata	

Table 6. Setting up HIDE ACTs permissions

- PROTECT: Providing SAS administrators group the write access to metadata.

Group	Grant	Deny
PUBLIC		WriteMetadata, WriteMemberMetadata, CheckInMetadata, Write, Administer
SAS Administrators	WriteMetadata, WriteMemberMetadata, CheckInMetadata, Write, Administer, ReadMetadata	

Table 7. Setting up PROTECT ACTs permissions

- LIMITDATA: Limiting the data availability accessed through information maps, OLAP server or metadata LIBNAME engine.

Group	Grant	Deny
PUBLIC		Read

Table 8. Setting up LIMITDATA ACTs permissions

The mutually exclusive and restricted access to designated groups towards certain metadata folders are depicted in a plan (refer to below table) using the 3 ACTs and specific explicit grants.

We start putting access controls from the highest folder in metadata folder hierarchy. The ACTs are put first to limit data and then grants are put to allow only a limited set of actions over the folders and objects.

The table below, shows folder hierarchy for internal contents, distributed across different departments and common (shared content across departments) work area in the development environment.

Folder	Direct Controls		Actions Affected and Limitations on Enforcement
	ACTs (LIMIT)	**Explicit Grants (ALLOW)**	
Environment (SAS VA Metadata Root Folder)	PROTECT + LIMITDATA	SASUSERS: +RM, +R SYSTEM ADMIN: +WMM, +WM, +CIM, +W, +A, +RM METADATA & DATA ADMIN: +WMM, +WM, +CIM, +W, +A, +RM USER: +RM, +R	All below folders (created after settings are applied) inherit same grant accesses for SASUSERS, SAS Administrators, System and Metadata Administrators.
- ORG	HIDE	Developers: +RM, +R	Below folders inherit USER group's read access
- - Department 1		Developer – Group 1: +WMM, -WM	Developer – Group 1 (content creators) only will have the write access
- - - Data			Write is disabled at this folder
- - - - Input Data		Developer – Group 1: +WM	Write metadata access is given for explicit write access to child objects
- - - - Output Data			Write is disabled at this folder
- - - - - LASR Data		Developer – Group 1: +WM	Write metadata access is given for explicit write access to child objects
- - - - - HDFS Data		Developer – Group 1: +WM	Write metadata access is given for explicit write access to child objects
- - - - Library		Developer – Group 1: +WM	Write metadata access is given for explicit write access to child objects
- - - - Autoload		Developers: -RM, -R	Only Administrators will have access to put tables in this folder (schedule)
- - - - - Append			
- - - - - Unload			
- - - - - Format			
- - - - - Log			
- - - - Data Query		Developer – Group 1: +WM	Write metadata access is given for explicit write access to child objects
- - - - Reports		Developer – Group 1: +WM	Write metadata access is given for explicit write access to child objects
- - - - Explorations		Developer – Group 1: +WM	Write metadata access is given for explicit write access to child objects
- - - - Jobs		Developer – Group 1: +WM	Write metadata access is given for explicit write access to child objects
- - Department 2		Developer – Group 2: +WMM, -WM	Similar structure and permissions as Department 1
- - Department 3		Developer – Group 3: +WMM, -WM	Similar structure and permissions as Department 1
- - Common		Developer – Group 1: +WMM, -WM Developer – Group 2: +WMM, -WM Developer – Group 3: +WMM, -WM	Additional permissions for each child folders since content creators for Department 1, 2 and 3 groups will have write access in Common folder

Table 9. Metadata folder permissions for organization users' (internal)

The table below, explains the access controls over the folder structure designated for the public users (external to the organization) view.

Folder	Direct Controls		Actions Affected and Limitations on Enforcement
	ACTs (LIMIT)	**Explicit Grants (ALLOW)**	
- GLOBAL	HIDE	Global Users: +RM, +R	Below folders inherit Public group's read access
- - Department 1		Developer – Group 1: +WMM, -WM	Developer – Group 1 only will have the write access (same structure like above)
- - Department 2		Developer – Group 2: +WMM, -WM	Similar structure and permissions as Department 1
- - Department 3		Developer – Group 3: +WMM, -WM	Similar structure and permissions as

			Department 1
- - Common		Developer – Group 1: +WMM, -WM Developer – Group 2: +WMM, -WM Developer – Group 3: +WMM, -WM	Additional permissions for each child folders since Department 1, 2 and 3 groups will have write access in Common folder

Table 10. Metadata folder permissions for global users (external)

The explicit grants are suggested to convert into ACTs. These ACTs should be configured with explicit groups (not SASUSERS or PUBLIC) as best practice such that granting access is easy. Deny operations are already setup with the general-use base ACTs. In this paper, all SASUSERS group members needed to get read access in the development environment.

BEST PRACTICES IN METADATA ARCHITECTURE DESIGN

ACTs should be protected too from anyone other than administrators to read and edit them. It can be restricted through SAS Management Console through Authorization tab. It is recommended to keep the following accesses over these ACTs.

- System Admin: +RM, +WM
- Metadata & Data Admin: +RM, +WM
- SAS Administrators: +RM, +WM
- SAS System Services: +RM, -WM
- PUBLIC: -RM, -WM
- SASUSERS: +RM, -WM

Administrators should have “grant” access where you put SASUSERS have “deny”. Otherwise, it will “deny” Administrators too.

Don’t create the metadata folder structure at one go without applying the ACTs and Explicit Grant Accesses. This prevents the inheritance of security to child folders and objects. Administrators should follow a sample flow provided below.

a) Create one parent folder inside repository
b) Apply necessary ACTs
c) Apply Explicit Grants
d) Continue again from step (a) for all child folders

WMM should be applied inside at the highest parent folder level (Department 1 etc.) to enable writing metadata. WM should be applied at the objects level to provide write metadata access.

General Public group can have all the other content consumers (Users) as members, such that they can consume any globally available report. But this would be a system administrator’s discretion.

SETTING UP THE VISUAL ANALYTICS ENVIRONMENT

Once we have documented all aspects of the metadata architecture and security rules, we implement the same and test, before the environment is opened for business users. Some aspects of environment setup are explained below.

CREATE CUSTOM ACT FILE

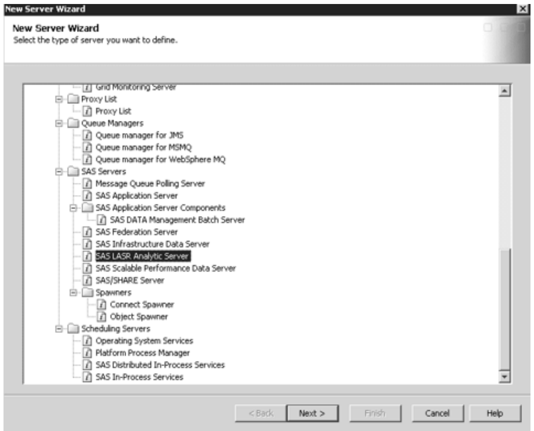
While implementing security, we can either export-import ACTs or have a script creating custom made ACTs for us. Below code uses “sas-make-act” batch tool to create a custom ACT called “Hide” that blocks the ReadMetadata permission for all groups except SAS Administrators and SAS System Services.

```
sas-make-act -profile Admin Hide -create -deny PUBLIC:RM
   -grant "SASAdministrators":RM -grant "SAS System Services":RM
```

SETTING UP THE DISTRIBUTED LASR SERVERS

For setting up the multiple LASR analytics servers in VA environment, administrators should login into SAS Management Console with LASRADM login ID and navigate to Server Manager.

Clicking on "New Server" provides a new server wizard with step-by-step configuration.

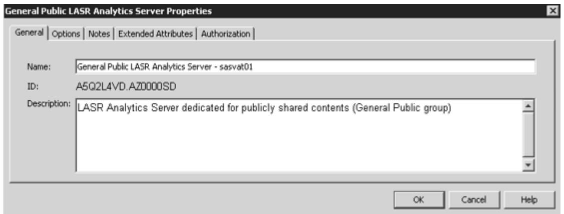


New Server Wizard

New Server Wizard
Select the type of server you want to define.

```
  Grid Monitoring Server
⊟ Proxy List
    Proxy List
⊟ Queue Managers
    Queue manager for JMS
    Queue manager for MSMQ
    Queue manager for WebSphere MQ
⊟ SAS Servers
    Message Queue Polling Server
    SAS Application Server
  ⊟ SAS Application Server Components
      SAS DATA Management Batch Server
    SAS Federation Server
    SAS Infrastructure Data Server
    SAS LASR Analytic Server
    SAS Scalable Performance Data Server
    SAS/SHARE Server
  ⊟ Spawners
      Connect Spawner
      Object Spawner
⊟ Scheduling Servers
    Operating System Services
    Platform Process Manager
    SAS Distributed In-Process Services
    SAS In-Process Services
```

Figure 7. Create a New Server wizard with server type options



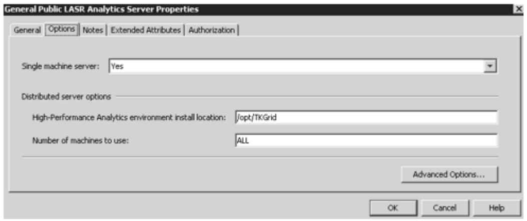
General Public LASR Analytics Server Properties

Name: General Public LASR Analytics Server - sasvat01

ID: A5Q2L4VD.AZ0000SD

Description: LASR Analytics Server dedicated for publicly shared contents (General Public group)

Figure 8. Server name and properties setup



General Public LASR Analytics Server Properties

Single machine server: Yes

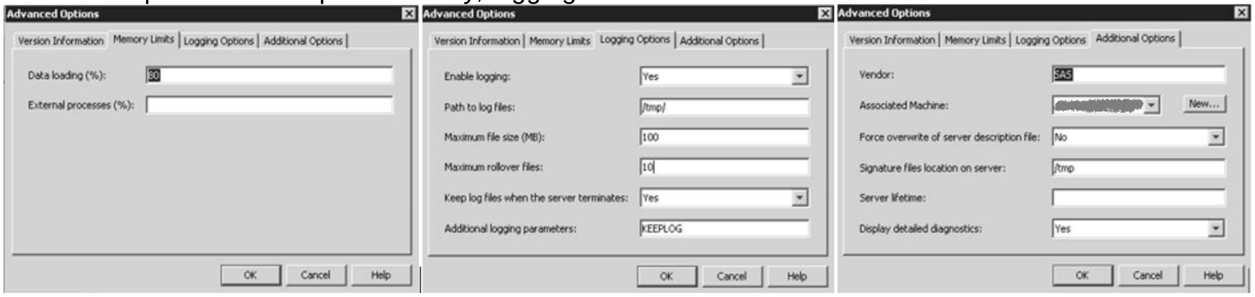
Distributed server options

High-Performance Analytics environment install location: /opt/TKGrid

Number of machines to use: ALL

Figure 9. Server type (distributed / non-distributed)

Advanced options are setup for memory, logging and server connection.



Advanced Options

Data loading (%): 80

External processes (%):

Advanced Options

Enable logging: Yes

Path to log files: /tmp/

Maximum file size (MB): 100

Maximum rollover files: 10

Keep log files when the server terminates: Yes

Additional logging parameters: KEEPLOG

Advanced Options

Vendor: SAS

Associated Machine: [redacted]

Force overwrite of server description file: No

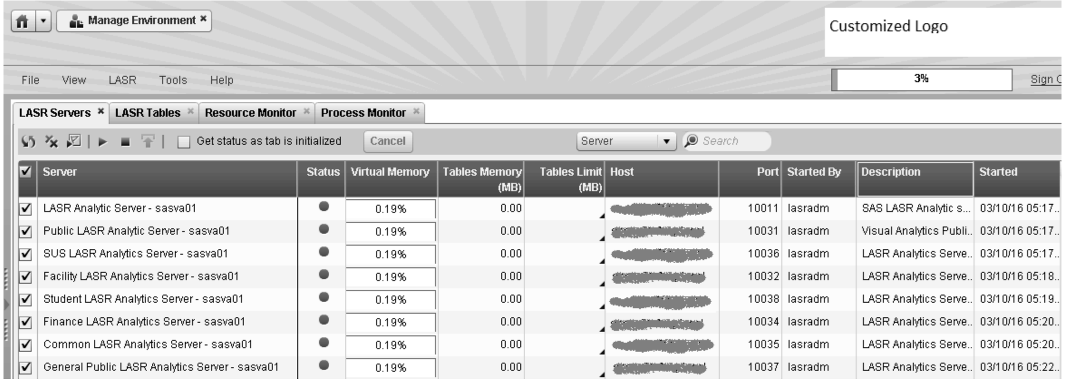
Signature files location on server: /tmp

Server lifetime:

Display detailed diagnostics: Yes

Figure 10. Advanced options setup for the LASR Server

Once the LASR analytics servers are setup in distributed environment, they can be managed through the Environment Manager under administrator login. As a best practice LASRADM ID should be used to manage these servers.



☑	Server	Status	Virtual Memory	Tables Memory (MB)	Tables Limit (MB)	Host	Port	Started By	Description	Started
☑	LASR Analytic Server - sasva01	●	0.19%	0.00		[redacted]	10011	lasradm	SAS LASR Analytic s...	03/10/16 05:17..
☑	Public LASR Analytic Server - sasva01	●	0.19%	0.00		[redacted]	10031	lasradm	Visual Analytics Publi..	03/10/16 05:17..
☑	SUS LASR Analytics Server - sasva01	●	0.19%	0.00		[redacted]	10036	lasradm	LASR Analytics Serve..	03/10/16 05:17..
☑	Facility LASR Analytics Server - sasva01	●	0.19%	0.00		[redacted]	10032	lasradm	LASR Analytics Serve..	03/10/16 05:18..
☑	Student LASR Analytics Server - sasva01	●	0.19%	0.00		[redacted]	10038	lasradm	LASR Analytics Serve..	03/10/16 05:19..
☑	Finance LASR Analytics Server - sasva01	●	0.19%	0.00		[redacted]	10034	lasradm	LASR Analytics Serve..	03/10/16 05:20..
☑	Common LASR Analytics Server - sasva01	●	0.19%	0.00		[redacted]	10035	lasradm	LASR Analytics Serve..	03/10/16 05:20..
☑	General Public LASR Analytics Server - sasva01	●	0.19%	0.00		[redacted]	10037	lasradm	LASR Analytics Serve..	03/10/16 05:22..

Figure 11. Viewing all LASR servers in Environment Manager

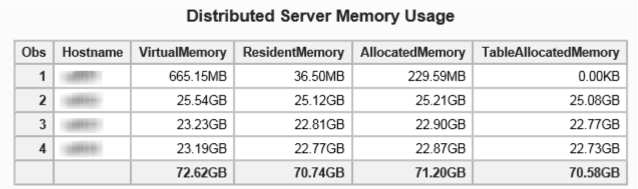
MONITORING LASR SERVER MEMORY

_T_LASRMEMORY table in SAS LASR Analytics server 2.5 contains details about server and table memory usage. This table is created dynamically when accessed through SASIOLA engine.

Below codes provides memory related details from the distributed LASR servers:

```
libname example sasiola host="grid001.example.com" port=10010 tag=hps;
data distributed;
  set example._T_LASRMEMORY;
run;
proc print data=distributed;
  title "Distributed Server Memory Usage";
  format _numeric_ sizekmg9.2;
  sum    _numeric_;
run;
```

Output of the program looks like:



Distributed Server Memory Usage

Obs	Hostname	VirtualMemory	ResidentMemory	AllocatedMemory	TableAllocatedMemory
1	[redacted]	665.15MB	36.50MB	229.59MB	0.00KB
2	[redacted]	25.54GB	25.12GB	25.21GB	25.08GB
3	[redacted]	23.23GB	22.81GB	22.90GB	22.77GB
4	[redacted]	23.19GB	22.77GB	22.87GB	22.73GB
		72.62GB	70.74GB	71.20GB	70.58GB

Figure 12. Code output - LASR memory usage

Similarly, the _T_TABLEMEMORY table provides details about memory used by tables. The number of records, in-memory size, and compressed / uncompressed size are some of the captured metrics.

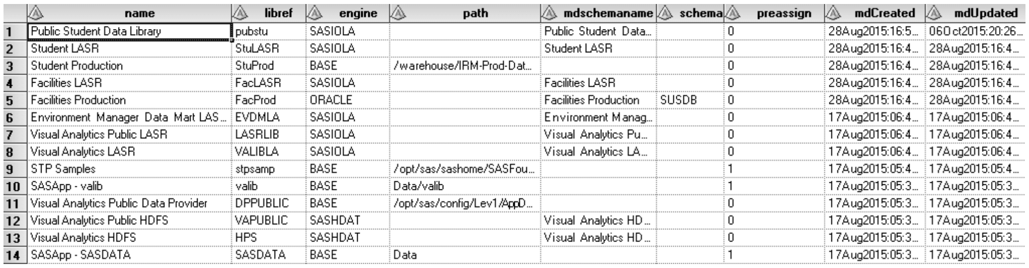
QUERYING METADATA

Pre and Post implementation of metadata objects, testing for confirmation is a good practice to check whether the path, engine, author and other credentials are in sync or not. Here is a base SAS code which provides details for metadata libraries in Foundation. You can change the same to capture details for reports (“report” is a metadata object type) as well.

```
Options metaserver="localhost" metaport=8561 metauser="sasadm@saspw" metapass="***"
metarepository="Foundation" metaprotocol=BRIDGE;

/* Begin query to metadata server for SAS Library objects */
data metadata_libraries;
length liburi upasnuri $256 name $128 type id $17
libref engine $8 path mdschemaname schema $256
preassign $ 1 mdCreated mdUpdated $ 18;
keep name libref engine path mdschemaname schema preassign mdCreated mdUpdated;
call missing(liburi,upasnuri,name,engine,libref,preassign,mdCreated,mdUpdated);

/* Get each Library object */
nlibobj=1;
librc=metadata_getnobj("omsobj:SASLibrary?@Id contains '.'",nlibobj,liburi);

/* For each library, retrieve the libref, engine, path */
     do while (librc>0);
          /* Get Library attributes */
          rc=metadata_getattr(liburi,'Name',name);
          rc=metadata_getattr(liburi,'Engine',engine);
          rc=metadata_getattr(liburi,'Libref',libref);
          rc=metadata_getattr(liburi,'IsPreassigned',preassign);
          rc=metadata_getattr(liburi,'MetadataCreated',mdCreated);
          rc=metadata_getattr(liburi,'MetadataUpdated',mdUpdated);
```

```
         /* Get the Directory and DatabaseSchema object associated with this library via a
         UsingPackages association*/
         n=1;
         uprc=metadata_getnasn(liburi,'UsingPackages',n,upasnuri);
         /* We found a UsingPackages association */
         if uprc > 0 then do;
            /* Determine object type */
            call missing(type,id,path,mdschemaname,schema);
            rc=metadata_resolve(upasnuri,type,id);
            if type='Directory' then do;
                   /* Get the path and output the record */
                   rc=metadata_getattr(upasnuri,'DirectoryName',path);
                   output;
            end; /*if type='Directory'*/
            else if type='DatabaseSchema' then do;
                   /* Get the schema and output the record */
                   rc=metadata_getattr(upasnuri,'Name',mdschemaname);
                   rc=metadata_getattr(upasnuri,'SchemaName',schema);
                   output;
            end; /*if type='DatabaseSchema'*/

            /*Check to see if there are any more Directory objects*/
            n+1;
            uprc=metadata_getnasn(liburi,'UsingPackages',n,upasnuri);
         end; /*if uprc > 0*/

         /*Look for another library*/
         nlibobj+1;
         librc=metadata_getnobj("omsobj:SASLibrary?@Id contains '.'",nlibobj,liburi);
    end; /*do while (librc>0)*/
run;
```

The output looks like below table.



	name	libref	engine	path	mdschemaname	schema	preassign	mdCreated	mdUpdated
1	Public Student Data Library	pubstu	SASIOLA		Public Student Data...		0	28Aug2015:16:5...	06Oct2015:20:26...
2	Student LASR	StuLASR	SASIOLA		Student LASR		0	28Aug2015:16:4...	28Aug2015:16:4...
3	Student Production	StuProd	BASE	/warehouse/IRM-Prod-Dat...			0	28Aug2015:16:4...	28Aug2015:16:4...
4	Facilities LASR	FacLASR	SASIOLA		Facilities LASR		0	28Aug2015:16:4...	28Aug2015:16:4...
5	Facilities Production	FacProd	ORACLE		Facilities Production	SUSDB	0	28Aug2015:16:4...	28Aug2015:16:4...
6	Environment Manager Data Mart LAS...	EVDMLA	SASIOLA		Environment Manag...		0	17Aug2015:06:4...	17Aug2015:06:4...
7	Visual Analytics Public LASR	LASRLIB	SASIOLA		Visual Analytics Pu...		0	17Aug2015:06:4...	17Aug2015:06:4...
8	Visual Analytics LASR	VALIBLA	SASIOLA		Visual Analytics LA...		0	17Aug2015:06:4...	17Aug2015:06:4...
9	STP Samples	stpsamp	BASE	/opt/sas/sashome/SASFou...			1	17Aug2015:05:4...	17Aug2015:05:4...
10	SASApp - valib	valib	BASE	Data/valib			1	17Aug2015:05:3...	17Aug2015:05:3...
11	Visual Analytics Public Data Provider	DPPUBLIC	BASE	/opt/sas/config/Lev1/AppD...			0	17Aug2015:05:3...	17Aug2015:05:3...
12	Visual Analytics Public HDFS	VAPUBLIC	SASHDAT		Visual Analytics HD...		0	17Aug2015:05:3...	17Aug2015:05:3...
13	Visual Analytics HDFS	HPS	SASHDAT		Visual Analytics HD...		0	17Aug2015:05:3...	17Aug2015:05:3...
14	SASApp - SASDATA	SASDATA	BASE	Data			1	17Aug2015:05:3...	17Aug2015:05:3...

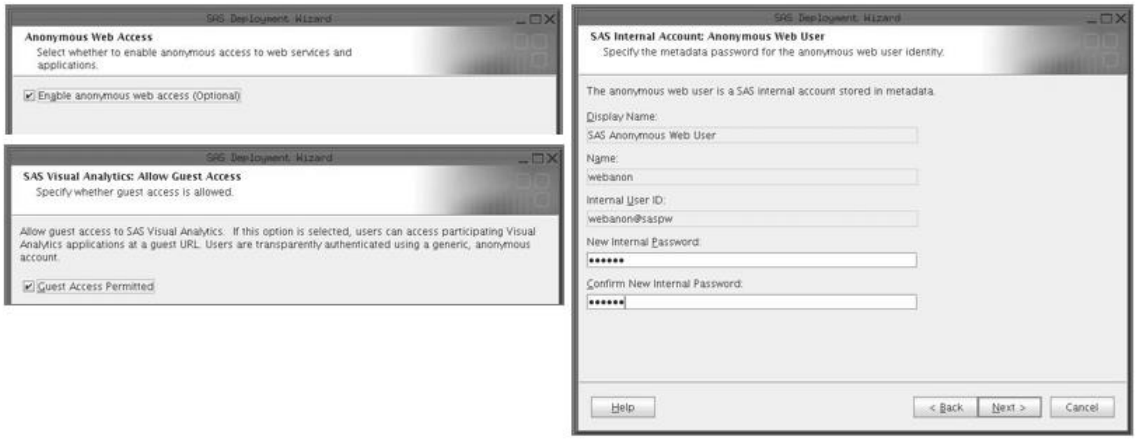
Figure 13. Metadata library listing through query

GUEST ACCESS

Having a guest access to Visual Analytics contents through VA hub is an optional feature that allows anonymous access to a limited number of Visual Analytics resources and functions. Visual Analytics guest access allows users without a metadata or OS account to view reports and resources that are setup by administrators. Guest access link to Hub is *http://host:7980/SASVisualAnalyticsHub/guest.jsp*.

Users of the guest access URL authenticates using the SAS Anonymous Web user (webanon) account. This is a SAS internal account and there is no authentication ability from an individual host or directory server account.

Guest access can be enabled during middle-tier configuration. To enable it, select the check box to enable Anonymous Web Access, specify a password for the account and select the check box “Guest Access Permitted” in SAS Deployment Wizard.



Anonymous Web Access
Select whether to enable anonymous access to web services and applications.

☑ Enable anonymous web access (Optional)

SAS Visual Analytics: Allow Guest Access
Specify whether guest access is allowed.

Allow guest access to SAS Visual Analytics. If this option is selected, users can access participating Visual Analytics applications at a guest URL. Users are transparently authenticated using a generic, anonymous account.

☑ Guest Access Permitted

SAS Internal Account: Anonymous Web User
Specify the metadata password for the anonymous web user identity.

The anonymous web user is a SAS internal account stored in metadata.

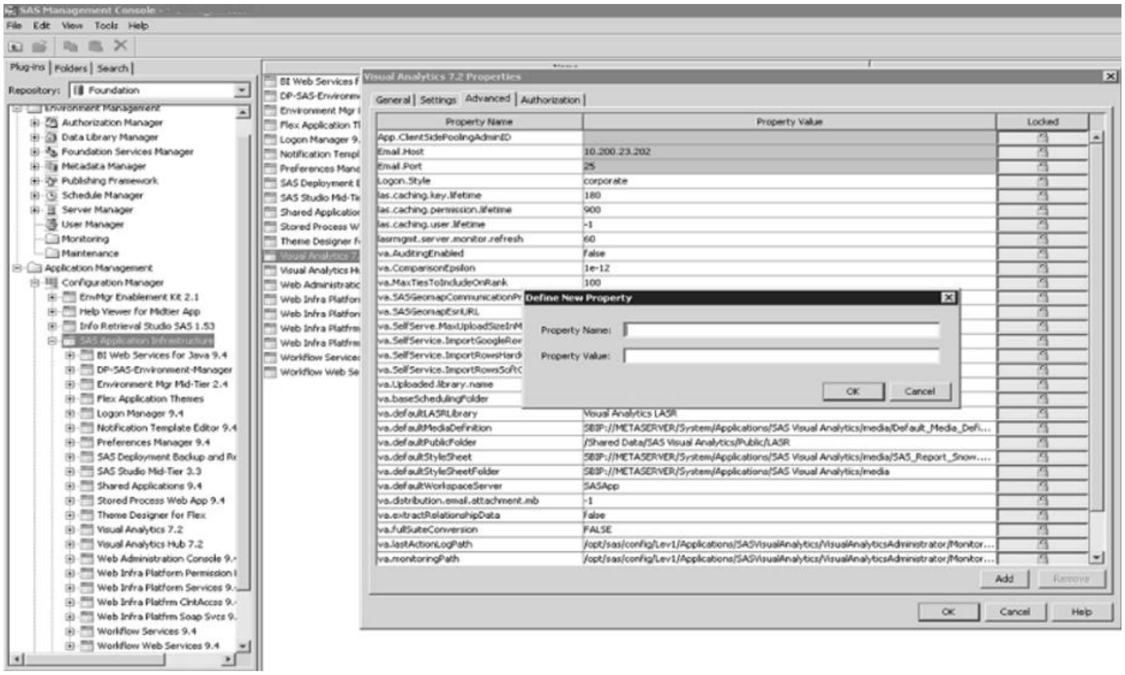
Display Name: SAS Anonymous Web User
Name: webanon
Internal User ID: webanon@saspw
New Internal Password: ••••••
Confirm New Internal Password: ••••••

Figure 14. Enabling Guest Access during middle-tier configuration

Guest access can be disabled permanently through SAS Management Console or temporarily by disabling SAS Anonymous Web User account. Below are the steps for enabling guest users to view shared contents in Visual Analytics Hub.

1) Log into the SAS Management Console as the SAS Administrator User
2) Expand Plug-ins > Application Management > Configuration Manager > SAS Application Infrastructure
3) Right-click on Visual Analytics 7.2 and select Properties
4) Select the Advanced tab
5) Double-click Property Value field for “App.AllowGuest”;
   ```
   Enter “true” to enable and “false” to disable
   ```
6) Click OK to close the Visual Analytics 7.2 Properties window and save the change
7) Restart the web application servers (steps for Linux)
   ```
   <SAS_CONFIG_ROOT>/Lev1/Web/WebAppServer/SASServer1_x/bin/tcruntime-ctl.sh restart
   ```
8) Restart the web application server (steps for Windows)
   a. From the Windows Start menu, type “services.msc” in the search window
   b. Open the Services application indicated by an icon with gears
   c. From the list of services, find the web application service, right-click and select Restart

Below diagram shows the interface from SAS Management Console.



Property Name	Property Value	Locked
App.ClientSidePoolingAdminID		
Email.Host	10.200.23.202	
Email.Port	25	
Logon.Style	corporate	
las.caching.key.lifetime	180	
las.caching.permission.lifetime	900	
las.caching.user.lifetime	-1	
lasrmgmt.server.monitor.refresh	60	
va.AuditingEnabled	false	
va.ComparisonEpsilon	1e-12	
va.MaxTiesToIncludeOnRank	100	
va.SASGeomapCommunicationPr		
va.SASGeomapEsriURL		
va.SelfServe.MaxUploadSizeInM		
va.SelfService.ImportGoogleRov		
va.SelfService.ImportRowsHard		
va.SelfService.ImportRowsSoftC		
va.Uploaded.library.name		
va.baseSchedulingFolder		
va.defaultLASRLibrary	Visual Analytics LASR	
va.defaultMediaDefinition	SBIP://METASERVER/System/Applications/SAS Visual Analytics/media/Default_Media_Defi...	
va.defaultPublicFolder	/Shared Data/SAS Visual Analytics/Public/LASR	
va.defaultStyleSheet	SBIP://METASERVER/System/Applications/SAS Visual Analytics/media/SAS_Report_Snow....	
va.defaultStyleSheetFolder	SBIP://METASERVER/System/Applications/SAS Visual Analytics/media	
va.defaultWorkspaceServer	SASApp	
va.distribution.email.attachment.mb	-1	
va.extractRelationshipData	false	
va.fullSuiteConversion	FALSE	
va.lastActionLogPath	/opt/sas/config/Lev1/Applications/SASVisualAnalytics/VisualAnalyticsAdministrator/Monitor...	
va.monitoringPath	/opt/sas/config/Lev1/Applications/SASVisualAnalytics/VisualAnalyticsAdministrator/Monitor...	

Figure 15. The SAS Management Console interface to allow Guest user access

From the guest URL, if you are prompted for credentials, then Anonymous access is disabled. If instead, you are logged in automatically, it confirms that anonymous access was properly enabled.

For a report to be visible via Guest Access, the SAS Anonymous Web User should have Read Metadata and Read data access either as himself or implicitly via SASUSERS or PUBLIC groups' permissions inheritance.

DATA MANAGEMENT THROUGH "AUTOLOAD" OPTION

Autoload provides the capability to periodically synchronize in-memory data against tables in a designated folder. You can use Autoload to keep a set of source tables in memory ready-to-use state when you need it. Users or processes can place source tables in a specified host location, called a drop zone. Corresponding in-memory data is periodically updated to reflect the contents of the drop zone.

Autoload periodically scans the contents of the designated drop zone, which is referred to as the Autoload data directory. After each scan, Autoload synchronizes in-memory data against the source tables in Autoload data directory. Autoload checks the file timestamps to make a decision on syncing.

- Tables that are already in memory will be refreshed if the source table has been updated.
- If the source tables are not already in memory they will be automatically loaded based on the Autoload schedule.
- Tables that's are newer than in-memory, are unloaded and reloaded.
- Tables placed at "Unload" folder, are unloaded from memory during Autoload schedule.
- Source tables that are in the "Append" subdirectory and newer than their corresponding in-memory tables are appended to their corresponding in-memory tables. If a table in the Append subdirectory has no corresponding in-memory table, it is loaded into a new table.

Advantages of Autoload are:

- No need to start and stop the LASR servers
- No metadata registration is required for source tables
- Browser based file import size constraints are not applicable

The public Autoload LASR server and library are created during configuration. The standard Autoload configuration comes with the below artifacts:

- Autoload data directory (drop-zone): /AppData/SASVisualAnalytics/VisualAnalyticsAdministrator/AutoLoad
- Scripts directory: /Applications/SASVisualAnalytics/VisualAnalyticsAdministrator/
- LASR Server: Public LASR Analytic Server
- Library: Visual Analytics Public LASR
- Library Metadata Folder: /Shared Data/SAS Visual Analytics/Public/LASR
- Default scheduling option: For Unix a cron job is scheduled to run every 15 minutes (editable)
- Formats: used to load user defined formats during Autoload

Below are example steps to configure and enable Autoload into the standard application directory.

1) Change ownership of the default directory structure to the LASR administrator account (typically LASRADM).

```
        > cd <SAS_CONFIG_ROOT>/Lev1/Applications/SASVisualAnalytics/
        > chown -R lasradm:sasusers VisualAnalyticsAdministrator
        > ls –al VisualAnalyticsAdministrator
```

2) As the LASR administrator account, execute the schedule.sh script. This script does not echo messages to the screen.

```
        > cd VisualAnalyticsAdministrator
        > ./schedule.sh
```

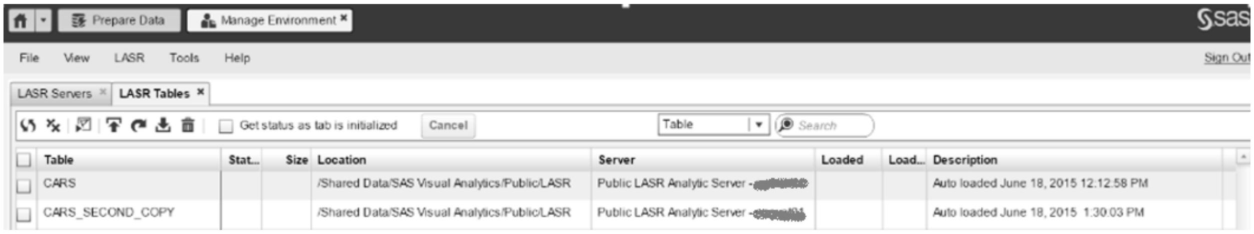
3) To verify that it ran properly, check the cron table for this account. The script should have created an entry that runs every 15 minutes.

```
        > crontab -l
        > */15 * * * * /opt/sas/config/Lev1/Applications/SASVisualAnalytics
/VisualAnalyticsAdministrator/runsas.sh > /dev/null
```

In addition to standard Public Autoload setup, private and customized Autoload libraries and servers can be created through below steps.

1) Create a new Autoload data directory as follows:
   - autoload-data-branch/new_folder
   - autoload-data-branch/new_folder/Append
   - autoload-data-branch/new_folder/Formats
   - autoload-data-branch/new_folder/Logs
   - autoload-data-branch/new_folder/Unload
2) Create a new Autoload scripts directory.
3) In the new autoload-scripts-branch/new_folder directory, edit the copied files as follows:
   - AutoLoad.sas
   - runsas.sh (or runsas.bat)
   - schedule.sh (or schedule.bat) and unschedule.sh (or unschedule.bat)
4) In SAS Management Console, identify or create a metadata folder for generated LASR table objects (in this example, /Shared Data/SAS Visual Analytics/Autoload/new_folder).
5) In SAS Management Console, configure a LASR library to support Autoload and add extended attributes.

Refer to the below screenshot for viewing 2 tables loaded in memory via Autoload option.



□	Table	Stat...	Size	Location	Server	Loaded	Load...	Description
□	CARS			/Shared Data/SAS Visual Analytics/Public/LASR	Public LASR Analytic Server - [redacted]			Auto loaded June 18, 2015 12:12:58 PM
□	CARS_SECOND_COPY			/Shared Data/SAS Visual Analytics/Public/LASR	Public LASR Analytic Server - [redacted]			Auto loaded June 18, 2015 1:30:03 PM

Figure 16. Two tables loaded via Autoload option

SETUP RELOAD-ON-START

While a user initiates an import of an XLS file, VA places a data set copy of the source data in the data provider library used as backing store. SAS loads the data from the backing store and creates a corresponding LASR table object. When the LASR server is stopped, the table is unloaded from in-memory. When the server is started from the LASR Servers tab, the data is reloaded from the backing store. In SAS Management console, options tab of a LASR server provides adding a data provider library (a base SAS library) which works as the backing store of the file. The extended attributes "VA.ReloadOnStart.Enabled" should be set to "YES" to enable Reload-on-start.

HOME PAGE CUSTOMIZATION FOR USERS

There are 2 customization properties for the home page, which allow or restrict making changes to home page for a user.

- home.allow.anonymous.user.personalization

  This property determines whether the service account that supports guest access can modify the home page. Valid values are "true" and "false". The default is "false".

- sas.home.exclude.capability.users

  This property blocks individualized home page capabilities for one or more users regardless of group and role memberships.

  Default value is "SAS Anonymous Web User". To block, leave it unspecified. To block a specific user, specify the user ID that is stored for that user in metadata. To block multiple users, specify a comma-separated list of user IDs. To unblock all users, leave this property unspecified and set the property sas.home.allow.anonymous.user.personalization to "true".

Additionally, to restrict home page setting allowed only for administrators, remove "Home: Usage" and "Home: Administration" from SASUSERS group roles and add "Home: Usage" and "Home: Administration" into roles for groups SAS Admins, Metadata Admin Group, and System Admin Group.

OPTIMIZE REPORTING PERFORMANCE THROUGH FULLCOPIES

If frequently used reports are based on smaller tables (less than 2GB), consider FullCopies option over the non-distributed environment to get better performance, as data is consolidated. To optimize retrieval performance for smaller tables, a distributed SAS LASR Analytic Server can keep multiple consolidated (full non-distributed) copies of each table. Each copy is written to and retrieved from a single machine. Each machine launches its own non-distributed server processes as needed to fulfill load and access requests. Load balancing and reuse of the non-distributed server processes further enhance performance.

To optimize high-volume access to smaller tables in a distributed SAS LASR Analytic Server, identify or create a LASR library that is exclusively for smaller tables. Associate the library with a distributed SAS LASR Analytic Server and set the attribute "VA.TableFullCopies" to a positive integer (usually less than 4) in LASR library's Extended Attributes.

MONITORING RESOURCES

Administrators have some high-level platform performance monitoring capability through Environment Manager, Administrator Console in Hub.

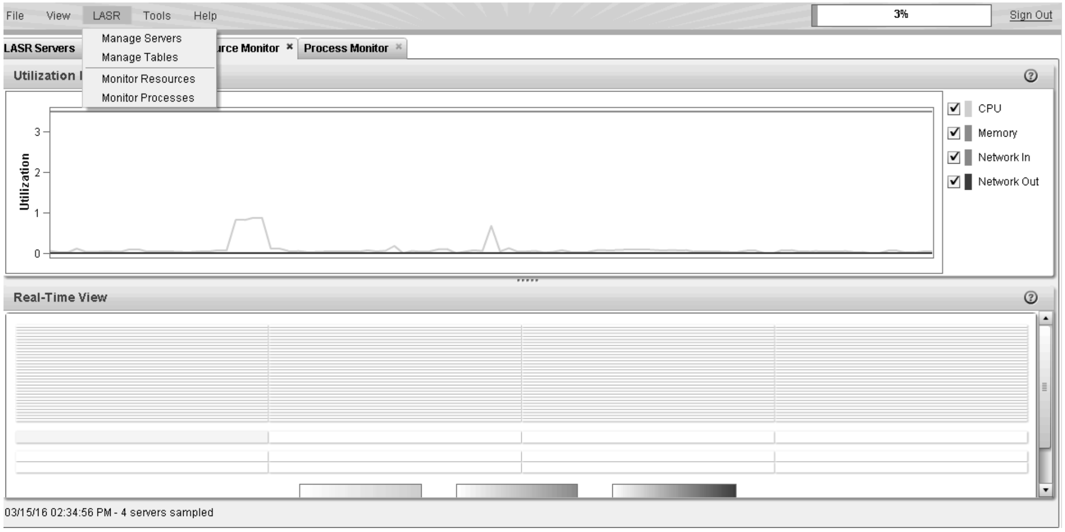


Figure 17. Administrator reporting and monitoring capability in Admin console in Hub

CUSTOMIZED THEMES

Administrators can build customized Flex themes in administrator console (Theme Designer) and deploy them to be the default theme for Visual Analytics Hub applicable for all users.

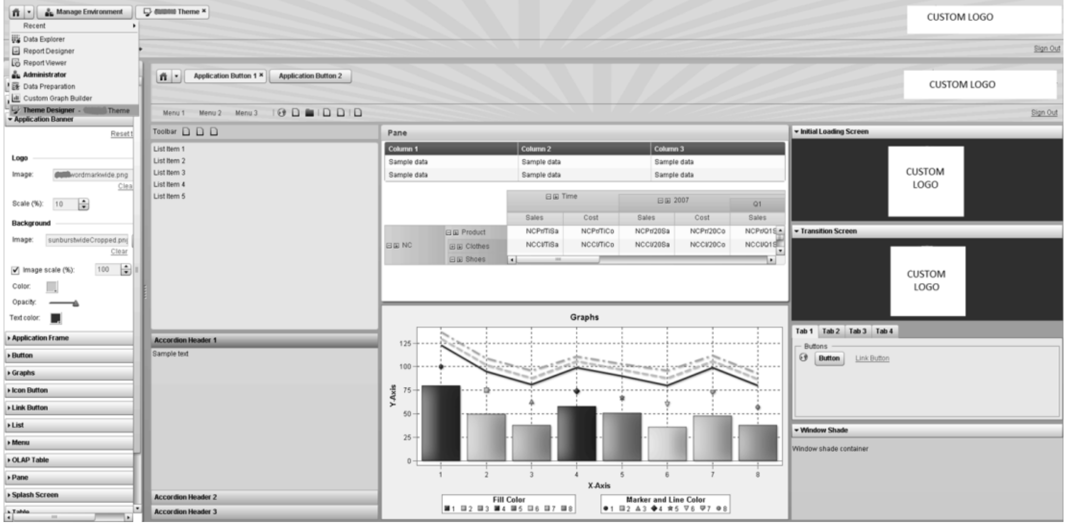


Figure 18. Theme designer to design custom themes

CONCLUSION

Structured metadata streamlines the analytical and reporting process, as well as contributes to controlled implementation. The advanced configuration and robust architecture helps Administrators to keep the content creation and consumption process less cluttered and easy to manage. Metadata and content migration becomes easy over multiple environments. The advanced administrative capabilities help build standardized themes and look-and-feel across environments. Advanced data handling techniques improve the data availability and thus lead to faster analytics and decision making. Administrators now have more power (advanced Admin dashboard) and less worry (easily knowing where to look when error happens). That leads to more doing, bigger innovations and better customer experience.

REFERENCES

- SAS® Visual Analytics 7.2: Administration Guide, SAS
- SAS® 9.4 Intelligence Platform Security Administration Guide, Second Edition, SAS
- Cecily Hoffritz and Johannes Jørgensen, SAS Institute Inc., Copenhagen, Denmark. “Best Practice Implementation of SAS® Metadata Security at Customer Sites in Denmark”. Paper 376-2011. SAS Global Forum 2011.
- SAS Visual Analytics User Guide - http://support.sas.com/documentation/onlinedoc/va/

ACKNOWLEDGMENTS

We thank KAVI Global for providing us with all the necessary infrastructure and resources. We also wanted to thank Arun Sugumar from KAVI Global for his valuable suggestions and guidance. We are also grateful to SAS for giving us this opportunity to present and publish our work and learnings.

CONTACT INFORMATION

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

Ratul Saha
KAVI Global
(847) 387-6760
ratul.saha@kaviglobal.com
http://www.kaviglobal.com/

Vimal Raj
KAVI Global
(847) 387-6760
vimal.raj@kaviglobal.com
http://www.kaviglobal.com/

Vignesh Balasubramanian
KAVI Global
(847) 387-6760
vignesh.balasubramanian@kaviglobal.com
http://www.kaviglobal.com/