Paper 311-2010

A Practical Approach to Securing a SAS® 9.2 Intelligence Platform Deployment

Jim Fenton, SAS Institute, Inc., Denver, CO
Robert Ladd, SAS Institute, Inc., Phoenix, AZ

ABSTRACT

The SAS 9.2® Intelligence Platform is a multi-tiered environment consisting of components residing on client desktops, middle-tier Web servers, computing servers, and (most importantly) secured data assets. This multi-tiered design offers tremendous flexibility and configurability but requires careful planning and deployment to provide a secure working environment. This document is a practical example of a secure SAS Intelligence Platform deployment, based on actual customer requirements, providing different user groups access to various secured data assets, compute server capabilities, desktop client and Web-based application functionality. This document is intended as a guide for SAS administrators and assumes that you are familiar with the concepts and terminology introduced in the *SAS® 9.2 Intelligence Platform: Security Administration Guide*.

INTRODUCTION

When it comes to SAS 9.2 metadata and security, each SAS site is unique, with different needs and priorities. This is especially true at a granular level, where different strategies are required to meet business needs and IT standards while keeping the environment secure. A common goal is to create a flexible, extendable, and robust model while staying administratively friendly. Often, however, metadata security models are designed to encompass everything and become difficult to use and too complicated to administer.

There's a fine line between excessive design and not doing enough. Just because you can do something doesn't mean it's the best thing to do. An important step in customizing the security model is to have a clear understanding of the problem to solve along with a good foundation in the SAS metadata security concepts. Thorough planning can help bridge the gap between a successful security model and one that is not secure.

To build a successful SAS metadata security model, you must know (or have a good idea) what needs to be secured, understand the default SAS metadata security model, and then know how to modify this default model to fulfill the specific security requirements. The example presented in this paper is just one way to design a security model. This paper discusses foundations for developing a successful model, the process for developing a security model, and the method for implementing the security model.

FOUNDATIONS FOR SECURITY MODEL DEVELOPMENT

In order to design and administer a successful metadata security model, it is critical to understand the following:

- the permission abbreviations
- the metadata authorization layer
- the rules or precedence within users and groups
- the authorization decision making process

These foundations are leveraged throughout this paper; therefore it's worth spending a few moments in review. For more detailed information, refer to the *SAS® 9.2 Intelligence Platform: Security Administration Guide*.

PERMISSION ABBREVIATIONS

RM	Read Metadata
WM	Write Metadata
WMM	Write Member Metadata
CM	Check-In Metadata
A	Administer
R	Read
C	Create
W	Write
D	Delete

METADATA AUTHORIZATION LAYER

Figure 1 represents the hierarchy of metadata permissions applied to objects.

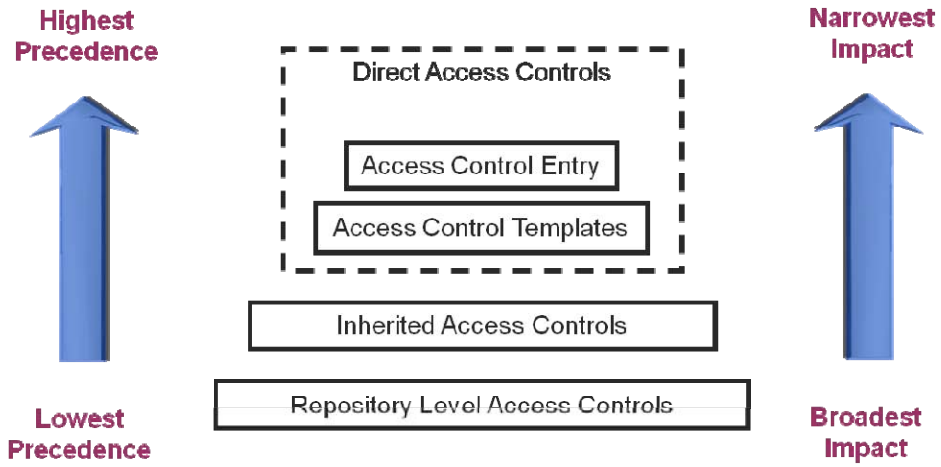

Figure 1. Metadata Authorization Layer

From top to bottom, the access controls are ordered as follows:

- from highest precedence (hardest to override) to lowest precedence (easiest to override)
- from narrowest impact (specific object) to broadest impact (many objects)

For example, a repository-level access control template (ACT) can affect all resources in a repository, whereas an access control entry (ACE) can be assigned to only one resource.

A permission setting that is a direct access control for one resource (such as a report folder) can be the source of an inherited permission for another resource (such as a report within that folder).

RULES OF PRECEDENCE WITHIN USERS AND GROUPS

Figure 2 represents the user and group identity hierarchy within the SAS metadata.

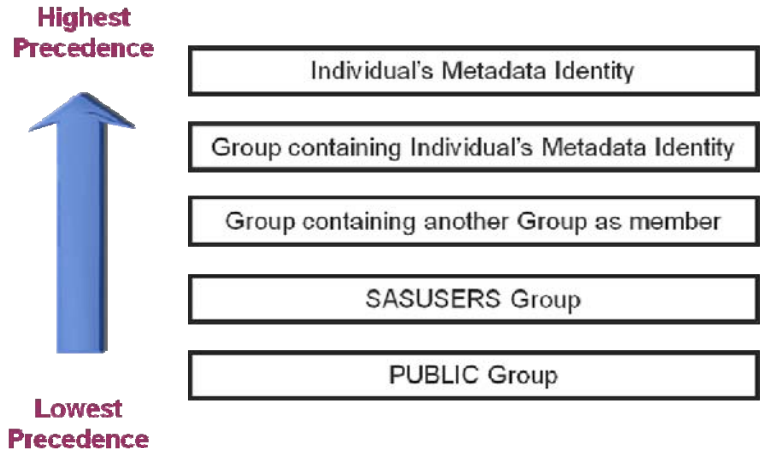

Figure 2. User and Group Identity Hierarchy

The following examples illustrate how the identity hierarchy works.

User Information	User's Identity Hierarchy
User has no metadata identity.	Primary identity: Unknown First-level membership: PUBLIC
User has a metadata identity and no explicit group memberships.	Primary identity: Self First-level membership: SASUSERS Second-level membership: PUBLIC
User is a direct member of two user-defined groups (Group B Users and XCMD Users).	Primary identity: Self First-level memberships: Group B Users, XCMD Users Second-level membership: SASUSERS Third-level membership: PUBLIC
User is a direct member of a user-defined group (Group A Users), and this group is a member of another group (The Group A Users group is a member of the SASApp Server Users group).	Primary identity: Self First-level membership: Group A Users Second-level membership: SASApp Server Users Third-level membership: SASUSERS Fourth-level membership: PUBLIC

Table 1. Examples of User and Group Identity Hierarchy

AUTHORIZATION DECISION-MAKING PROCESS

The following list describes the authorization decision process:

1. Permissions set on the target resource are examined.
   - Permissions determined through group membership are resolved by the identity hierarchy. For example, a permission that is assigned to a user overrides a different permission that is assigned to a group to which the user belongs.
   - If there is an ACE and an ACT at the same level in the identity hierarchy, the permissions in the ACE take precedence.
   - If there are two ACTs with different permissions on the same object, the outcome is a denial.
   - If there are no pertinent permissions set on the target resource, then the evaluation process continues.

2. The inheritance rules are applied to identify the target resource's parent objects. Although it is rare to have more than one parent in SAS 9.2 metadata, the process is applied to each parent object.
   - If **any** parent objects convey a grant, access is granted.
   - If **all** parent objects convey denials, access is denied.
   - If the target resource does not have any parent objects, then the evaluation process continues.

3. The **Users and Permissions** tab in the repository ACT is examined. Any conflicts within the repository ACT are resolved by the identity hierarchy.
   - If the repository ACT grants or denies the requested permission, that grant or denial is determinative.
   - If the repository ACT neither grants nor denies the permission, the permission is denied.

Note: If there is no repository ACT, the permission is granted. There should always be a designated repository ACT.

The process flow in Figure 3 details the authorization decision process.

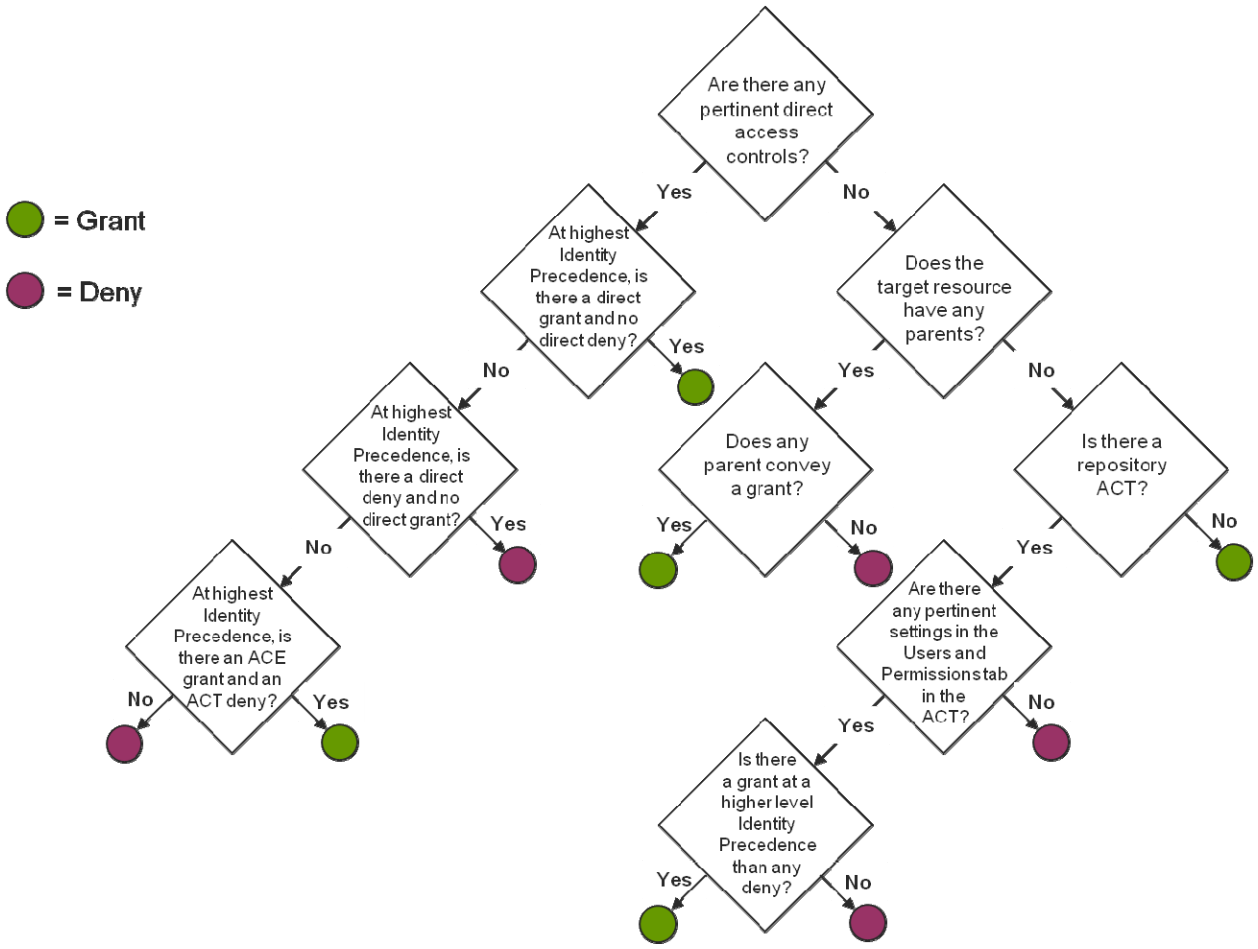

Figure 3. Authorization Decision Process

Table 2 summarizes the precedence principles for metadata-layer access controls and presents an example of each principle.

Principles of Metadata-Layer Access Control Precedence		
Principle	**Example**	
	Scenario	**Outcome and Explanation**
Permissions set on an object have precedence over permissions on the object's parent.	The permissions set on the Group A folder (Group A ACT) deny RM to SASUSERS. The permissions set on the SAS Folders folder grant RM to SASUSERS.	Users that are not granted RM permission in Group A ACT cannot see the Group A Folder. Direct access controls always have precedence over inherited controls, regardless of who the permissions are assigned to.
Conflicting permissions on an object are resolved using identity	SASUSERS are denied RM permission on the Group A folder. The Group A Users group is granted RM	Group A Users can see the Group A folder. The Group A Users group (contains the individual's metadata identity) is a higher-

Principles of Metadata-Layer Access Control Precedence		
Principle	**Example**	
	Scenario	**Outcome and Explanation**
precedence.	permission on the Group A folder.	level precedence than the SASUSERS group.
A grant from any parent will provide access to the object.	The SASApp – Logical Stored Process server has no ACT or direct access permissions applied to it. The SASApp Server ACT is applied to the parent SASApp server context.	All members of Group A and Group C have access to the SASApp stored process server, but Group B members do not. All permissions for the SASApp stored process server are inherited from the permissions applied to the SASApp server context.
If there are no relevant direct controls and there are no parent objects, then the repository ACT determines the outcome.	There is no custom ACT applied to the SASApp server context or SASApp servers.	In the initial deployment, everyone with a metadata identity has access to the SASApp servers. The SASUSERS group has access to the SASApp servers through permissions granted in the Default ACT.

Table 2. Principles of Metadata-Layer Access Control Precedence

PROCESS FOR DEVELOPING A SECURITY MODEL

When you build a SAS metadata security model, there are three basic metadata objects used to secure content: groups, access control templates (ACT), and server contexts. Although users play a part in the security model as members in groups, they are too granular for most security applications. The example presented in this paper uses groups, ACTs, and server contexts within the SAS Intelligence Platform. The idea is to illustrate different approaches and minimize the number of objects created, maintained, and secured in SAS metadata.

The first step in developing a SAS metadata security model is to define what needs to be secured in the SAS environment. A best practice is to pick patterns of access requirements and design a template using these patterns. This template can be applied to other groups as they fit these patterns, or new patterns can be designed. The key is having a model that is flexible, extendable, and simple to administer.

REQUIREMENTS

The example used throughout this paper restricts some user groups to a single server context while providing unique access to file system resources. It also illustrates how to restrict access to specialized server contexts while providing access to unique servers within the server context.

The security requirements for this environment are defined below:

1. There are three user groups: Groups A, B, and C.
2. Groups A and C contain administrators, content developers, and casual users. Administrators are developers who have access to SAS® Management Console for group administrative tasks. Developers create content in SAS® Enterprise Guide projects, stored processes, OLAP cubes, information maps, and SAS® Web Report Studio reports for users. Users have no ability to create metadata content outside their own personal metadata folders.
3. Group B contains administrators and power users. Administrators are power users who have access to SAS Management Console for group administrative tasks. Power users create content in SAS® Enterprise Guide projects and stored processes.
4. There are two Power User subgroups in Group B: those who need access to system commands and those who do not.

5. There is no data or metadata content shared between groups.
6. Group administrator tasks are defined as creating and managing libraries, table server content, user and group assignments, group folders, publishing channels, and job schedules.
7. This is a secure environment and *no one* can save data from the server environment to a local machine. The only exception is reports that users create. These reports can be only printed and not saved locally. Data can be imported from local desktops to the SAS environment, but once on the server it remains in the server environment.
8. There is some metadata content that must be hidden from everyone except the SAS Administrator and the SAS Demo User. Although this is not the norm in a new SAS metadata environment, there might be a need to hide objects from everyone but the SAS Administrator and a small set of users.
9. Only users with a metadata identity can access the SAS environment.

After the security requirements are defined, the next step is to design a security model template that fulfills these requirements. The two main resources to secure are application servers and metadata content folders. The following section will review the default security model for SAS Folders, describe how to develop a security model template for the content folders, and then leverage this model in designing a server security template.

THE DEFAULT SAS METADATA SECURITY MODEL

A custom security model integrates with the default security model that exists in the SAS metadata environment. There are ACT and ACE modifications applied to metadata objects that must be understood in order to prepare the environment to support custom security models.

The entry level access control template is the Default ACT (repository ACT). It defines the baseline access controls for SAS metadata and requires modifications to support a secure environment. A key item to note is that inheritance occurs within an ACT based on the precedence rules for users and groups. This means that the SASUSERS group inherits permissions applied to the PUBLIC group and all other groups inherit the SASUSERS group permissions by default inside an ACT. Taking this into consideration, the necessary Default ACT modifications are as follows:

1. The preceding requirements state that only users with a metadata identity can access the SAS environment. To comply with this requirement, the PUBLIC group must be explicitly denied all permissions in the Default ACT.
2. After PUBLIC is explicitly denied, access must be granted to those who have a metadata identity (SASUSERS). This group is added and granted explicit RM and WM permissions. The Write Metadata (WM) permission is required to establish back-end metadata associations, like creating and associating a user's personal My Folder directory to the user when initially accessing the SAS environment. Write Member Metadata is a subset of Write Metadata (WM), is inherited from WM, and must be explicitly denied so that SASUSERS do not have open access to the folder level. Also, SASUSERS do not require R access to data because at the level that this ACT is applied there is no requirement to read data. Custom ACTS will be created and applied for this purpose.
3. The SAS System Services group (sastrust) and SAS Administrators group (sasadmin@saspw) are in the Default ACT at initial deployment. The SAS System Services group requires explicit RM and WM permissions. Write Metadata is required to create metadata associations on behalf of the user.
4. The SAS Administrators group needs explicit RM, WM, CM, and A permissions. This group manages the metadata environment, does not have access to data or the file system, and does not require a grant of R, W, C, or D.
5. The SAS General Servers group (sassrv), which runs the stored process servers, must be added and explicitly granted RM and R permission, and explicitly denied the WM permission inherited from SASUSERS. Read (R) permission is needed to access data used in stored processes.

Figure 4 illustrates the modified Default ACT permissions. The gray shaded boxes are inherited permissions from other groups within this ACT. The letter codes G and D stand for Grant and Deny permissions.

ACT Desc & Members	Actual ACT Settings								
	RM	WM	WMM	CM	R	W	C	D	A
Default ACT									
SAS General Servers (sassrv)	G	D	D	D	G	D	D	D	D
SAS System Services (sastrust)	G	G	D	D	D	D	D	D	D
SAS Administrators	G	G	D	G	D	D	D	D	G
SASUSERS	G	G	D	D	D	D	D	D	D
PUBLIC	D	D	D	D	D	D	D	D	D

Figure 4. Default ACT Template

When securing metadata folders, there is another baseline ACT that must be considered. The SAS Administrator Settings ACT is applied by default to the SAS Folders folder and directly affects the permissions applied by the Default ACT. In the initial deployment, only the SAS Administrators and SAS System Services groups are included, with applied grants as shown in Figure 5. This is sufficient and there is no need to alter this ACT.

ACT Desc & Members	Actual ACT Settings								
	RM	WM	WMM	CM	R	W	C	D	A
SAS Administrator Settings									
SAS System Services (sastrust)	G								
SAS Administrators	G	G		G					G

Figure 5. SAS Administrators Settings ACT Template

One last item to consider with SAS folder security is the access control entry. In the initial deployment, PUBLIC is explicitly denied WM and CM in the SAS Folders ACE. If the SAS Administrator Settings ACT was not applied to this folder, this explicit change would deny the SAS Administrator's group WM and CM permission through inheritance from PUBLIC within this ACE. These explicit denials affect the SASUSERS group that is granted RM and WM in the Default ACT. With PUBLIC denied WM in this ACE, which has a higher-level precedence than the Default ACT, SASUSERS inherit this denial and now have only RM permission on SAS Folders and all child folders.

Figure 6 illustrates the effective folder permissions on SAS Folders and most children. Notice that the applied ACT permissions are green, inherited Default ACT permissions are gray, and the explicit ACE denials for PUBLIC are white. All child folders represent these permissions as inherited gray. The WMM grant for SAS Administrators is inherited from the WM grant within this ACT.

ACT Desc & Members	Effective Folder Permissions								
	RM	WM	WMM	CM	R	W	C	D	A
Default ACT									
SAS General Servers (sassrv)	G	D	D	D	G	D	D	D	D
SAS System Services (sastrust)	G	D	D	D	D	D	D	D	D
SAS Administrators	G	**G**	**G**	G	D	D	D	D	G
SASUSERS	G	D	D	D	D	D	D	D	D
PUBLIC	D	D	D	D	D	D	D	D	D

Figure 6. Default ACT Effective Folder Permissions

CREATING CUSTOM ACTS

At this point, everyone who has a metadata identity has access to all the metadata folders through the SASUSERS permissions. Custom group ACTs must be created to modify these permissions and restrict access to the group's resources. Starting with Group A, the ACT for this group would contain the following permission changes:

- SASUSERS are denied RM and WM. This denial stops everyone's access. Because the SASUSERS group is at a lower-level precedence than the groups added below, the permissions granted to these groups will precede this explicit denial pattern and provide the appropriate access where this ACT is applied.
- Grant SAS Administrators RM, WM, CM, and A.

- Grant SAS System Services (sastrust) RM.
- Grant SAS General Servers (sassrv) RM and R.
- Add the group Group A Administrators and grant RM, WM, R, W, and A.
- Add the group Group A Developers and grant RM, WM, R, and W.
- Add the group Group A Users and grant RM.

The Group A ACT permissions pattern looks like the Figure 7 template. The inherited WM denials come from the explicit denial on SASUSERS. This permission pattern gives the Group A Administrators the permission to create content and administer the environment. Group A Developers create content and Group A Users consume this content. Only groups granted permission will have access to the content where this ACT is applied.

ACT Desc & Members	Actual ACT Settings								
	RM	WM	WMM	CM	R	W	C	D	A
Group A ACT									
SAS General Servers (sassrv)	G	D			G				
SAS System Services (sastrust)	G	D							
SAS Administrators	G	G		G					G
SASUSERS	D	D							
Group A Administrators	G	G			G	G			G
Group A Developers	G	G			G	G			
Group A Users	G	D			G				

Figure 7. Group A ACT Template

This ACT is ready to be applied to metadata content for Group A. Noting the requirements defined earlier, Group C shares the same permission pattern and can be created in the same manner using Group C groups.

When the Group A ACT is applied to the Group A folder, the effective permission pattern will look like Figure 8. The PUBLIC group comes from Default ACT inheritance. The bolded WMM inheritance is from the WM permissions.

ACT Desc & Members	Effective Folder Permissions								
	RM	WM	WMM	CM	R	W	C	D	A
Group A ACT									
SAS General Servers (sassrv)	G	D	D	D	G	D	D	D	D
SAS System Services (sastrust)	G	D	D	D	D	D	D	D	D
SAS Administrators	G	**G**	**G**	G	D	D	D	D	G
SASUSERS	D	D	D	D	D	D	D	D	D
PUBLIC	D	D	D	D	D	D	D	D	D
Group A Administrators	G	**G**	**G**	D	G	G	D	D	G
Group A Developers	G	**G**	**G**	D	G	G	D	D	D
Group A Users	G	D	D	D	G	D	D	D	D

Figure 8. Group A ACT Effective Folder Permissions

Next, the Group B ACT template is created. It is similar to the Group A ACT, except that Power Users take the place of the Developers. When applied to the Group B folder, this permission pattern grants Group B Administrators the permission to create content and administer the environment, while Group B Users create content. Figure 9 illustrates the Group B ACT.

ACT Desc & Members	Actual ACT Settings								
	RM	WM	WMM	CM	R	W	C	D	A
Group B ACT									
SAS General Servers (sassrv)	G	D			G				
SAS System Services (sastrust)	G	D							
SAS Administrators	G	G		G					G
SASUSERS	D	D							
Group B Administrators	G	G			G	G			G
Group B Users	G	G			G	G			

Figure 9. Group B ACT Template

The next two ACTs are for the subset of Group B users who need access to workspace servers enabled for operating system commands (XCMD). The names of the ACTs are based on the capability provided. These ACTs require additional metadata groups that divide the Group B users into the XCMD Users and NOXCMD Users. Figures 10 and 11 define these ACTs.

ACT Desc & Members	Actual ACT Settings								
	RM	WM	WMM	CM	R	W	C	D	A
XCMD ACT									
SAS General Servers (sassrv)	G	D							
SAS System Services (sastrust)	G	D							
SAS Administrators	G	G		G					G
SASUSERS	D	D							
XCMD Users	G	G							

Figure 10. XCMD ACT Template

ACT Desc & Members	Actual ACT Settings								
	RM	WM	WMM	CM	R	W	C	D	A
NOXCMD ACT									
SAS General Servers (sassrv)	G	D							
SAS System Services (sastrust)	G	D							
SAS Administrators	G	G		G					G
SASUSERS	D	D							
NOXCMD Users	G	G							

Figure 11. NOXCMD ACT Template

By now, a pattern is emerging where SASUSERS are leveraged to deny access to everyone and specific groups are granted permissions. This pattern is used again in the Hide ACT template in Figure 12. This security template is needed to hide content from everyone except the SAS Administrator and the SAS Demo User. Note that the SAS Demo User is granted WM permission as an example for providing granular user access to resources.

ACT Desc & Members	Actual ACT Settings								
	RM	WM	WMM	CM	R	W	C	D	A
Hide ACT									
SAS General Servers (sassrv)	G	D			G				
SAS System Services (sastrust)	G	D							
SAS Administrators	G	G		G					G
SASUSERS	D	D							
SAS Demo User	G	G			G				

Figure 12. Hide ACT Template

SECURING METADATA FOLDERS

The access control templates have been developed and are ready for application. The group metadata folders contain subfolders for organizing content. The ACT permissions applied at the parent group folder are inherited through all child-level folders and content. Figure 13 illustrates where the default and custom access control templates have been applied to metadata folders.

ACT Code	Folder	Comments
SASADM	SAS Folders	Parent folder
GrpA	Group A	
GrpB	Group B	
GrpC	Group C	
HIDE	1Development Area	Example of content to hide from users
HIDE	2Quality Assurance	Example of content to hide from users
HIDE	3Production Content	Example of content to hide from users
SASADM	Users	Parent to individual user folders

Figure 13. Metadata Folders Template

CREATING APPLICATION SERVERS FOR FILE SYSTEM ACCESS

Now that access control templates are in place to secure the metadata folders, the next step is to restrict access to system files from the SAS application servers. Each SAS server, except the OLAP server, has a file navigation definition that provides access to file system directories beginning at a defined point. This point could be the SAS User Root (for example, ..\My Documents\My SAS Files\9.2), System Root (for example, C:\), or a specific file system path. The SAS User Root option provides file system access, but by default is for the user and not shareable. The System Root, while shareable, depends solely on operating system security to protect resources. A best practice for securing resources is to define a specific path for each group in order to restrict access to defined file system resources. This requires physical server definitions in metadata for each group. The example in this paper defines specific SAS Workspace Servers. However, the process can be applied to pooled workspace servers and stored process servers.

The initial deployment configuration contains the SASApp server context, the SASApp – Logical Workspace Server definition and a SASApp – Workspace Server physical definition. The physical server definition is modified to define the file navigations for each group. To create a new physical server definition, refer to the topic, “Adding a New Server in an Existing Logical Server” in the *SAS 9.2 Intelligence Platform: Application Server Administration Guide*. In this example, two physical workspace server definitions are required under SASApp – Logical Workspace Server: Group A and Group C. Each server requires a unique name and port definition. A best practice is to define a template, as illustrated in Figure 19, for unique server port assignments and to design a pattern for growth.

When the servers are defined, file navigation on each physical server is set on the Advanced Options dialog box. The file navigations for the servers are defined in Figure 14.

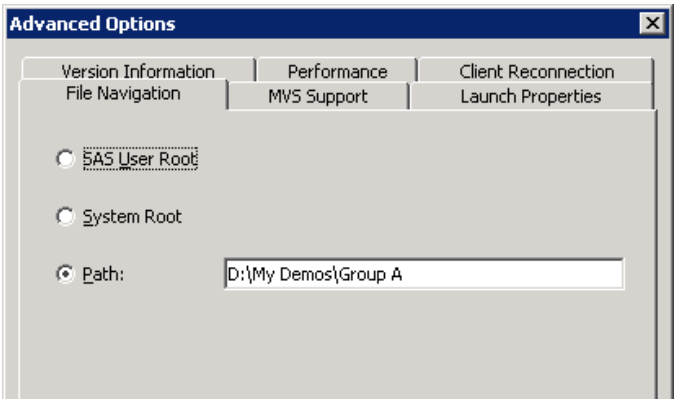

Figure 14. Workspace Server File Navigation Properties

Once complete, the ACTs are applied to the physical server definitions. Users should have access to *only one physical workspace definition per server context* to ensure consistent access to the same server definition and SAS environment.

Figure 15 illustrates the current workspace server configuration plan for Groups A and C.

ACT Code	Server Context	ACT Code	Server	Ports	File Navigation
	SASApp	**GrpA**	SASApp - Workspace Server - Grp A	8591	D:\Group A
		GrpC	SASApp - Workspace Server – Grp C	8592	D:\Group C
			SASApp - Stored Process Server	8601, 8611, 8621, 8631	D:\
			SASApp - OLAP Server	5451	
			SASApp - Pooled Workspace Server	8701	D:\

Figure 15. Server Context Template

The workspace server configuration for Group B looks similar to the pattern above, but differs in a couple of ways. Group B is on a separate server context (SASApp1) with the Group B ACT applied to this object. The physical server definitions contain the same file navigation path, but different server configuration options. One server is configured with the default NOXCMD option, and the second is configured to allow X system commands. The XCMD ACT and NOXCMD ACT are applied to the respective workspace server definitions assigning Group B members to the appropriate physical workspace definition.

To configure a physical server definition to allow XCMD, navigate to the Advanced Options dialog box for the server and select the **Allow XCMD** check box on the **Launch Properties** tab (Figure 16).

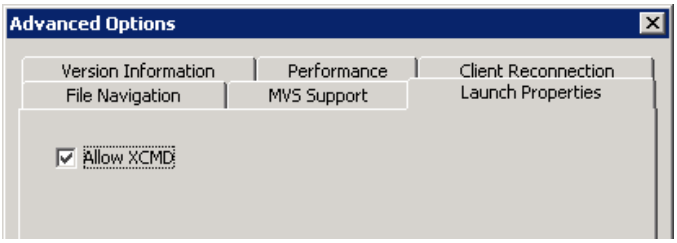

Figure 16. Workspace Server Launch Properties

Figure 17 illustrates the current workspace server configuration plan for Groups A, B, and C, as well as a pattern for growth.

ACT Code	Server Context	ACT Code	Server	Ports	File Navigation
	SASApp	GrpA	SASApp - Workspace Server - Grp A	8591	D:\Group A
		GrpC	SASApp - Workspace Server – Grp C	8592	D:\Group C
			SASApp - Stored Process Server	8601, 8611, 8621, 8631	D:\
			SASApp - OLAP Server	5451	
			SASApp - Pooled Workspace Server	8701	D:\
GrpB	SASApp1	NOXCMD	SASApp1 - Workspace Server – NOXCMD	8593	D:\Group B
		XCMD	SASApp1 - Workspace Server - XCMD	8594	D:\Group B
			SASApp1 - Stored Process Server	8602, 8622, 8623, 8624	D:\Group B
	SASxx		SASxx - Workspace Server	8594	
			SASxx - Stored Process Server	8603, 8632, 8633, 8634	
			SASApp - OLAP Server	5452	
			SASAPP - Pooled Workspace Server	8702	

Figure 17. Server Context Template

This is one way to secure access to Group B server resources. How the environment is secured is a matter of choice, taking into consideration tradeoffs of additional ACTs, server contexts, and physical servers.

VERIFYING ACCESS TO RESOURCES

In Figure 17, Group B users have access to the SASApp server context, but no physical servers. SAS Enterprise Guide and the SAS® Add-In for Microsoft Office expose server contexts where there is access to a physical server. In the Java clients such as SAS® Data Integration Studio, SAS® Information Map Studio, and SAS® OLAP Cube Studio, the user works with metadata first and a physical server at execution time. Group B users running these Java clients see the SASApp server context, but do not see Group A or Group C data, nor can they start a workspace on the SASApp server context. With the Group B ACT applied at the SASApp1 server context, Group A and Group C users are unaware of SASApp1 (all clients). This is a perfect example of why designing the initial security model using a template is important. The approach provides the ability to identify and resolve issues before applying the model.

A best practice is to secure the SASApp server context to groups that use this server (Group A and C). This is accomplished by creating one more ACT, the SASApp Server ACT (Figure 18). Up to this point, all ACTs created contained groups of users. This ACT leverages groups of groups. For example, the new SASApp Server Administrators group contains the groups Group A Administrators and Group C Administrators.

ACT Desc & Members	Effective Folder Permissions								
	RM	WM	WMM	CM	R	W	C	D	A
SASApp Server ACT									
SAS General Servers (sassrv)	G	D							
SAS System Services (sastrust)	G	D							
SAS Administrators	G	G		G					G
SASUSERS	D	D							
SASApp Server Administrators	G	G							
SASApp Server Developers	G	G							
SASApp Server Users	G	D							

Figure 18. SASApp Server ACT Template

Figure 19 illustrates the final workspace server configuration for Groups A, B, and C after the SASApp Server ACT is applied to the SASApp server context. The ACT permits access to only the logical and physical servers appropriate for each group.

ACT Code	Server Context	ACT Code	Server	Ports	File Navigation
SASApp	SASApp	**GrpA**	SASApp - Workspace Server - Grp A	8591	D:\Group A
		GrpC	SASApp - Workspace Server – Grp C	8592	D:\Group C
			SASApp - Stored Process Server	8601, 8611, 8621, 8631	D:\
			SASApp - OLAP Server	5451	
			SASApp - Pooled Workspace Server	8701	D:\
GrpB	SASApp1	**NOXCMD**	SASApp1 - Workspace Server – NOXCMD	8593	D:\Group B
		XCMD	SASApp1 - Workspace Server - XCMD	8594	D:\Group B
			SASApp1 - Stored Process Server	8602, 8622, 8623, 8624	D:\Group B
	SASxx		SASxx - Workspace Server	8594	
			SASxx - Stored Process Server	8603, 8632, 8633, 8634	
			SASApp - OLAP Server	5452	
			SASAPP - Pooled Workspace Server	8702	

Figure 19. Final Server Context Template

DEFINING ROLES TO LIMIT CAPABILITIES

New to SAS 9.2 is the ability to define roles that determine what a user can do within client interfaces. Roles do not restrict what resources a user has access to, but do limit what a user can do with the available resources. Roles contain a defined set of capabilities (or features) available in the user interfaces for SAS Add-In for Microsoft Office, SAS Enterprise Guide, SAS Management Console, and SAS Web Report Studio.

Roles are based on a grant-centric model, which means permissions cannot be assigned to a role. There are a number of predefined roles in SAS metadata that are sufficient for some SAS environments. However, the ability exists to expand the capability with custom roles. Due to the additive nature of roles, a best practice involves creating custom roles to define specific capabilities for user profiles. This provides a simple, focused point for determining the capabilities available to a group.

For the paper example, role definitions are needed for the SAS Group Administrators, SAS Developers, and SAS Users. Roles transcend groups, meaning that the SAS Developers role could be applied to the Group A Developers, Group C Developers, and Group B Users groups. These groups need to leverage the same application features.

Roles are a key tool for complying with the requirement that no server data can be saved locally or transmitted from the server except in report form. There are many individual roles that control availability to menu items, plug-ins, and buttons within these applications. Some roles are required to provide or reduce the availability of features fit for the user profiles and to comply with the server data restriction. These selected roles are defined in Figure 20, with either a G identifying availability or a blank indicating that the feature is not available to the user profile. Included in the list are the SAS Management Console roles highlighting the limited capabilities available to the SAS Group Administrators. The capabilities inside the SAS Add-In for Microsoft Office and SAS Enterprise Guide for transferring data from the server are not available to anyone. A complete capability list is provided in the Metadata Authorizations Template, available at http://support.sas.com/saspresents/311-2010.zip.

Application	Role Group	Role	SAS - Group Administrators	SAS - Developers	SAS-Users
SAS Add-In for Microsoft Office 4.2	Save or Distribute	Copy and Paste SAS Server Content			
SAS Enterprise Guide 4.2	Save or Distribute	Save Files to Local Computer			
		Copy and Paste SAS Server Content			
		Send Content to E-mail Recipient			
	Data	Download Data Files to PC			
SAS Management Console 9.2	General	Access Unregistered Plug-ins			
	Plug-ins	Application Monitor			
		Authorization Manager			
		Data Library Manager	G		
		Folder View	G		
		Foundation Services Manager			
		Metadata Manager			
		Publishing Framework	G		
		Schedule Manager	G		
		Server Manager			
		User Manager	G		
		Enterprise Miner			
		Deployment Tester			
		Map Service Manager			
		Forecast Server			
		SAS OLAP Server Monitor			
		Table Server Manager	G		
		Configuration Manager			

Figure 20. Selected Roles Template

Figure 21 identifies where these roles have been applied to metadata groups.

Users / Groups	Roles Applied
Group A Administrators	SAS - Administrators
Group A Developers	SAS - Developers
Group A Users	SAS - Users
Group B Administrators	SAS - Administrators
Group B Users	SAS - Developers
Group C Administrators	SAS - Administrators
Group C Developers	SAS - Developers
Group C Users	SAS - Users

Figure 21. Applied Roles Template

MATCHING REQUIREMENTS TO SAS METADATA CHANGES

Earlier, the example requirements were identified. Table 3 lists these requirements and how they were addressed using metadata security.

Requirement	Metadata Security Application
There are three user groups: Groups A, B, and C.	Created three metadata groups and access control templates for Group A, Group B, and Group C users.
Groups A and C contain administrators, content developers, and casual users. Administrators are developers who have access to the SAS Management Console for group administrative tasks. Developers create content in SAS Enterprise Guide projects, stored processes, OLAP cubes, information maps, and SAS Web Report Studio reports for users. Users have no ability to create metadata content outside their own personal metadata folders.	With X representing A or C, created three groups of users, Group X Administrators, Group X Developers, and Group X Users. Added these groups to the Group X ACT and applied metadata permissions to enable each group to accomplish the required tasks. For example, the Group X Users groups have only Read Metadata permission, which restricts their ability to create or modify metadata to their My Folder personal metadata folder.
Group B contains administrators and power users. Administrators are power users who have access to SAS Management Console for group administrative tasks. Power users create content in SAS Enterprise Guide projects and stored processes.	Created two groups of users, Group B Administrators and Group B Users. Added these groups to the Group B ACT and applied metadata permissions to enable the group to accomplish the required tasks. The Group B Users differ from the other user groups above in that they have Write Metadata permission to create content in the SAS metadata environment.
There are two Power User subgroups in Group B: those who need access to system commands and those who do not.	Created two groups, the NOXCMD and XCMD groups. Added Group B users to one of these groups based on their system command needs. Created two access control templates, NOXCMD ACT and XCMD ACT, and added the appropriate group to each ACT. Created two physical server definitions, one with the default NOXCMD option and the other with the XMCD option. The ACTs were applied to respective physical server definitions, restricting the users to only one of these physical servers.
There is no data or metadata content shared between groups.	Created SAS metadata folders for each group's content and applied the group's ACT to that parent folder, restricting access to only that group.
Group administrator tasks are defined as creating and managing libraries, table server content, user and group assignments, group folders, publishing channels, and job schedules.	Used roles to limit access to only the SAS Management Console plug-ins required for these administrative tasks.
This is a secure environment and *no one* can save data from the server environment to a local machine. The only exception is reports that users create. These reports can be only printed and not saved locally. Data can be imported from local desktops to the SAS environment, but once on the server it remains in the server environment.	Used roles to remove access to features within the user interfaces of SAS Add-In for Microsoft Office, SAS Enterprise Guide, and SAS Web Report Studio that allow users to save data locally or copy data out of the SAS server environment.
There is some metadata content that must be hidden from everyone except the SAS Administrator and the SAS Demo User. Although this is not the norm in a new SAS metadata environment, there might be a need to hide objects from everyone but the SAS Administrator and a small set of users.	Created the Hide ACT that denied Read Metadata access to everyone except the SAS Administrators group and the SAS Demo User.
Only users with a metadata identity can access the SAS environment.	Denied the PUBLIC group access to the SAS environment.

Table 3. SAS Security Requirements Addressed

IMPLEMENTING THE SECURITY MODEL

After the template is complete, users, groups, and roles are added to the SAS metadata using the User Manager plug-in in SAS Management Console. Users and roles are assigned to the appropriate groups. Access control templates are created using the Authorization Manager plug-in, and appropriate permissions are granted or denied based on the template patterns. Server contexts, application servers, and physical server definitions are created using the Server Manager plug-in, with ACTs applied as defined in the template. Figure 22 illustrates how the example SAS metadata environment defined in this paper looks in SAS Management Console.

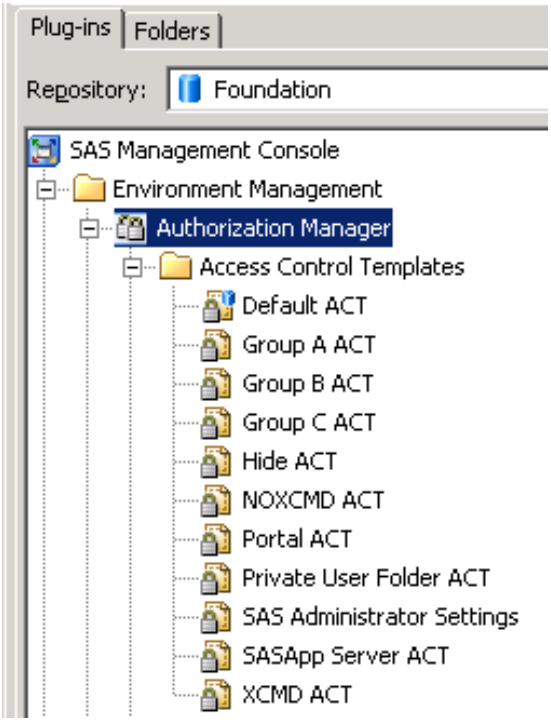

Figure 22. Authorization Manager Plug-In

The Access Control Templates list contains two ACTs created with the initial deployment in addition to those defined in this paper. The Portal ACT is a default template used to secure the SAS® Information Delivery Portal application metadata tree. The Private User Folder ACT is used to secure each user's My Folder folder. Both ACTs are beyond this paper's scope and are noted for information purposes. The Portal ACT is defined in the *SAS 9.2 Intelligence Platform: Web Application Administration Guide*, and information about the Private User Folder ACT can be found in the *SAS 9.2 Intelligence Platform: Security Administration Guide*.

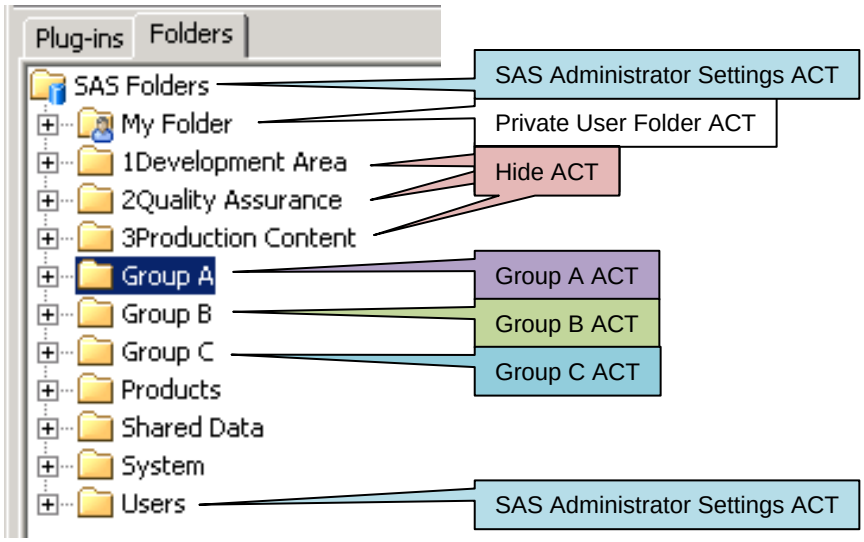

Figure 23. SAS Folders Plug-In

Figure 23 shows the parent content folders and the ACT applied to each folder. The effective permissions on the SAS Folders folder are listed in Figure 24.

ACT Desc & Members	Effective Folder Permissions								
	RM	WM	WMM	CM	R	W	C	D	A
SAS Folders folder									
SAS General Servers (sassrv)	G	D	D	D	G	D	D	D	D
SAS System Services (sastrust)	G	D	D	D	D	D	D	D	D
SAS Administrators	G	**G**	**G**	G	D	D	D	D	G
SASUSERS	G	D	D	D	D	D	D	D	D
PUBLIC	D	D	D	D	D	D	D	D	D

Figure 24. SAS Folders Effective Permissions

The effective permissions on the Group A folder are listed in Figure 25. Because the SASUSERS group is explicitly denied RM and WM in the ACT, any group that requires these permissions must be explicitly granted these permissions in the Group A ACT to precede the SASUSERS permissions. These explicit grants are seen in green. Any permission inheritance from the parent SAS Folders folder is preceded by the SASUSERS explicit permissions in the Group A ACT.

ACT Desc & Members	Effective Folder Permissions								
	RM	WM	WMM	CM	R	W	C	D	A
Group A ACT on Group A Folder									
SAS General Servers (sassrv)	G	D	D	D	G	D	D	D	D
SAS System Services (sastrust)	G	D	D	D	D	D	D	D	D
SAS Administrators	G	**G**	**G**	G	D	D	D	D	G
SASUSERS	D	D	D	D	D	D	D	D	D
PUBLIC	D	D	D	D	D	D	D	D	D
Group A Administrators	G	**G**	**G**	D	G	G	D	D	G
Group A Developers	G	**G**	**G**	D	G	G	D	D	D
Group A Users	G	D	D	D	G	D	D	D	D

Figure 25. Group A ACT Effective Permissions on Group A Folder

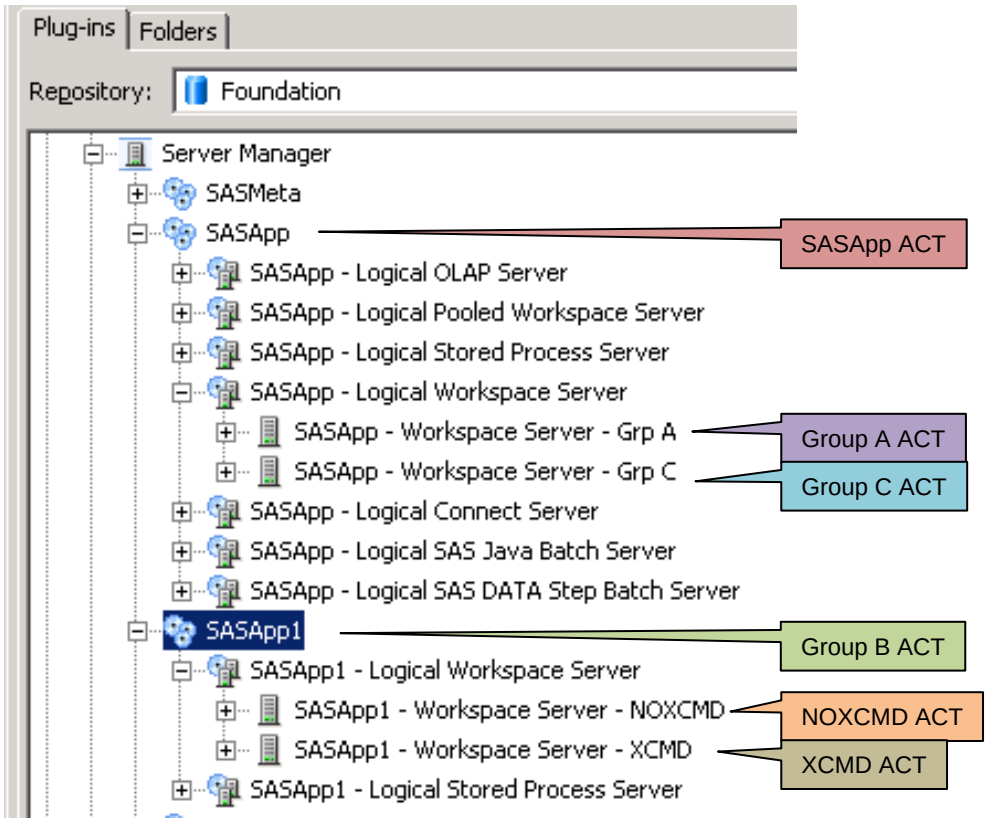

Figure 26. Server Manager Plug-In

Figure 26 lists application servers and applied ACTs. The effective permissions for SASApp1 are shown in Figure 27. The only permissions affecting servers are RM, WM, and A. All others are not present in the **Authorization** tab.

ACT Desc & Members	Effective Folder Permissions								
	RM	WM	WMM	CM	R	W	C	D	A
Group B ACT on SASApp1									
SAS General Servers (sassrv)	G	D	N/A	N/A	N/A	N/A	N/A	N/A	D
SAS System Services (sastrust)	G	D	N/A	N/A	N/A	N/A	N/A	N/A	D
SAS Administrators	G	G	N/A	N/A	N/A	N/A	N/A	N/A	G
SASUSERS	D	D	N/A	N/A	N/A	N/A	N/A	N/A	D
PUBLIC	D	D	N/A	N/A	N/A	N/A	N/A	N/A	D
Group B Administrators	G	G	N/A	N/A	N/A	N/A	N/A	N/A	G
Group B Users	G	G	N/A	N/A	N/A	N/A	N/A	N/A	D

Figure 27. Group B ACT Effective Permissions on SASApp1 Server Context

Figure 28 lists the effective permissions on the SASApp1 – Workspace Server – XCMD. The members include the Group B Administrators and Group B Users inherited from the Group B ACT applied to the parent SASApp1 server context. The permissions for these groups are inherited from SASUSERS in the XCMD ACT, which is at a higher precedence level explicitly denying RM and WM.

Notice that Group B Administrators are granted the A permission. SASUSERS are explicitly denied only RM and WM in the XCMD ACT that changes the effective permissions, but A is inherited through SASUSERS from the Default ACT, which is at a lower precedence level than the Group B ACT. This does not change the A permission. This could be corrected by explicitly denying A to SASUSERS in the XCMD ACT.

ACT Desc & Members	Effective Folder Permissions								
	RM	WM	WMM	CM	R	W	C	D	A
XCMD ACT on Workspace Server									
SAS General Servers (sassrv)	G	D	N/A	N/A	N/A	N/A	N/A	N/A	D
SAS System Services (sastrust)	G	D	N/A	N/A	N/A	N/A	N/A	N/A	D
SAS Administrators	G	G	N/A	N/A	N/A	N/A	N/A	N/A	G
SASUSERS	D	D	N/A	N/A	N/A	N/A	N/A	N/A	D
PUBLIC	D	D	N/A	N/A	N/A	N/A	N/A	N/A	D
Group B Administrators	D	D	N/A	N/A	N/A	N/A	N/A	N/A	**G**
Group B Users	D	D	N/A	N/A	N/A	N/A	N/A	N/A	D
XCMD Users	G	G	N/A	N/A	N/A	N/A	N/A	N/A	D

Figure 28. XCMD ACT Effective Permissions on the SASApp1- Workspace Server - XCMD

The final step is to validate the security model. This is accomplished by placing the test user ID into each group and validating that the group has access only to the appropriate server and metadata folder content resources. After this is completed, the SAS environment is secure and ready for use.

CONCLUSION

The approach used in this paper is an example of a process that has been successfully used to implement SAS metadata security models. The idea was to abstract what was learned during this implementation into a simple-to-understand process that can be adapted to most SAS metadata environments. As discussed, it is important to design an approach that is flexible, extendable, and simple to administer. Just because you can do something doesn't always mean that you should do it. In fact, the more complex the initial set-up, the more likely the model will be too restrictive and too difficult to maintain.

Taking into consideration that each SAS site has different requirements, there are options available to secure the environment. It is important to consider business needs and IT requirements when determining tradeoffs of additional groups, ACTs, server contexts, and servers. By having an iterative process to design patterns for implementing an extendable security model, you can easily make modifications while reducing time-consuming mistakes. The template approach provides not only a structured implementation plan, but a high-level picture of the security framework and thorough documentation of the model. With these tools, you have a practical approach to securing a SAS 9.2 Intelligence Platform deployment.

REFERENCES

SAS Institute Inc. 2009. *SAS 9.2 Intelligence Platform: Security Administration Guide.* Cary, NC: SAS Institute Inc. Available at http://support.sas.com/documentation.

SAS Institute Inc. 2006. *SAS 9.1.3 Intelligence Platform: System Administration Guide, Second Edition.* Cary, NC: SAS Institute Inc. Available at http://support.sas.com/documentation/configuration/bisecag.pdf.

SAS Institute Inc. 2009. *SAS 9.2 Management Console: Guide to Users and Permissions.* Cary, NC: SAS Institute Inc. Available at http://support.sas.com/documentation.

SAS Institute Inc. 2009. *SAS 9.2 Intelligence Platform: Web Application Administration Guide, Second Edition.* Cary, NC: SAS Institute Inc.
Available at http://support.sas.com/documentation.

SAS Institute Inc. 2009. *SAS 9.2 Intelligence Platform: Application Server Administration Guide.* Cary, NC: SAS Institute Inc. Available at http://support.sas.com/documentation.

ACKNOWLEDGMENTS

Many thanks to Diane Hatcher for her feedback and review of this paper.

RECOMMENDED READING

The most important information about Metadata Architecture and Security is in the *SAS 9.2 Intelligence Platform: Security Administration Guide*.

CONTACT INFORMATION

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

Jim Fenton
SAS Institute Inc.
Denver, CO 80401
Work Phone: (919) 531-9761
Fax: (919) 677-4444
E-mail: jim.fenton@sas.com
Web: www.sas.com

Robert Ladd
SAS Institute Inc.
Phoenix, AZ 85012
Work Phone: (602) 265-1616
Fax: (919) 677-4444
E-mail: robert.ladd@sas.com
Web: www.sas.com