Paper 094-2009

At the Controls – An Approach to Security via SAS® Management Console

Lisa Frank, BD, Sparks, MD

INTRODUCTION

BD, a leading global medical technology company that manufactures and sells medical devices, instrument systems and reagents, is dedicated to improving people's health throughout the world. BD is focused on improving drug therapy, enhancing the quality and speed of diagnosing infectious diseases, and advancing research and discovery of new drugs and vaccines. The Company's capabilities are instrumental in combating many of the world's most pressing diseases. Founded in 1897 and headquartered in Franklin Lakes, New Jersey, BD employs approximately 28,000 people in approximately 50 countries throughout the world. The Company serves healthcare institutions, life science researchers, clinical laboratories, industry and the general public.

What follows is a discussion regarding approaches to various aspects of metadata management as they relate to the management of security via SAS Management Console, based upon our experiences with the Enterprise Business Intelligence (EBI) Platform. Depending upon your environmental needs, not all of these approaches will work for you. However, it is hoped you may be able to glean one or two pieces of helpful information.

MANAGEMENT CONSOLE IN A NUTSHELL

SAS Management Console 9.1.3 is "a Java application that provides a single point of control for managing resources that are used throughout the Business Intelligence Architecture." (SAS) It is used by your SAS Administrator to manage server definitions, library definitions, user definitions, resource access controls, metadata repositories, etc.

PRIMARY USERS OF MANAGEMENT CONSOLE

It is strongly suggested access to the SAS Management Console product (M/C), be limited to the SAS Administrator and individuals designated as a backup for the SAS Administrator.

A TEAM APPROACH

You will find that a team approach is best when customizing M/C metadata permissions, to meet your EBI security needs. Therefore, if you don't already have one, establish a team to develop a road map on how to secure SAS components using M/C. Don't work in a vacuum. Seek advice from internally recognized security experts, regarding recommended approaches to security and to insure compliance with your company's policies. Do your home work. Be sure to tap into resources such as SAS white papers and user conferences, as there is much to be learned from the experiences of others.

DETERMINING SECURITY REQUIREMENTS FOR THE EBI PLATFORM TOOL SETS

Initially, the thought of establishing security requirements for the EBI Platform tool sets can be slightly overwhelming and a somewhat daunting task. It requires an understanding of each software component and how each component interacts and depends on other components.

In our case, members of the conversion team were responsible for exploring and developing implementation strategies for each of the EBI components. We established the infrastructure and corresponding security for Group Identities, User Identities, Access Control Templates (ACTs), Libraries, Stored Processes, Web Report Studio, Information Maps, SAS Information Delivery Portal and OLAP Cubes. Some areas were more involved than others, and the speed with which the strategies evolved and were finalized varied.

As the strategies and the security requirements for each of the EBI Platform Tool Sets and metadata structures were defined, test cases were established within M/C, to prove/disprove the feasibility of implementing the requirements. If deemed feasible, the necessary infrastructure, to support the security requirements for a given tool set, was established within M/C. If deemed not feasible, security requirements were revisited and alternatives explored. Initially, we did not realize the impact the security strategy for one component had on others. In some cases, we built and rebuilt security for the facets of M/C, multiple times, until we got it right.

You need to gain an understanding of how the security infrastructure for one component works with that of another, before you can begin to build your M/C infrastructure. Planning is critical. Consider the use of process maps to help define the individual component security strategies and how each affects the overall security strategy. It is important to establish a well thought out, solid approach for all components, before trying to build security prototypes within M/C.

For the remainder of this paper, we will discuss the various aspects of the metadata security. We will highlight some of the things our team learned through our implementation process, as well as, use an example to help illustrate a suggested approach for you to consider, when implementing the EBI Platform. Please keep in mind, the example is only one approach and its implementation depends upon your use of the EBI platform and the complexity of your organization.

USER TYPES

The foundation for any security strategy is users and groups. You need to determine what types of users to establish and what each type of user will be authorized to do. Therefore, it is suggested that you determine the types of users you need to define, early in your implementation process.

Let’s begin here with our example.

We will assume you have decided to categorize your users into five types and they are as follows:

1. Predefined SAS Users
2. Super Administrators
3. Administrators
4. Super Users
5. End Users

Predefined SAS Users are those identities delivered with the EBI Platform and should be utilized as recommended. Super Administrators and Administrators would be selected associates from your organization, responsible for the overall support and maintenance of the EBI Platform. Super Users and End Users would be members of your business community. The degree, to which each type of user can navigate within and throughout the EBI Platform would vary, with Super Administrators having the greatest power, followed by Administrators, Super Users and finally End Users.

Predefined SAS Users serve specific purposes within the EBI Platform, ranging from: having full permissions to maintain all metadata objects within the repository; to having specific permissions to administer the portal; to having restricted permissions allowing access only to testing new functionality.

Super Administrators would maintain and manage metadata definitions within M/C. They would utilize all available EBI tool sets, to develop and maintain applications, to be used by Super Users and End Users.

Administrators would utilize all available EBI tool sets, to develop and maintain applications, to be used by Super Users and End Users. They would not have any responsibility for or involvement in the maintenance or management of the metadata definitions within M/C.

Super Users would utilize Super User and End User applications via the SAS Add-In for Microsoft Office, SAS Enterprise Guide, the SAS Information Delivery Portal, etc. They would have the capability to develop SAS Web Report Studio reports for personal use, as well as, use by other authorized Super Users and End Users.

End Users would utilize End User applications via the SAS Add-In for Microsoft Office, SAS Enterprise Guide, the SAS Information Delivery Portal, etc. They would have the capability to develop SAS Web Report Studio reports for personal use only. However, they could execute SAS Web Report Studio reports developed by Super Users.

USER GROUPS

While you are not forced to establish user groups, they are highly recommended for organizational, as well as, ease of maintenance reasons.

As delivered, the SAS EBI Platform comes with a group named PUBLIC. Early in the implementation planning process, we chose to revoke all authority from the PUBLIC group, delivered by SAS. By revoking authority from the PUBLIC group, we gained complete control of who is granted access to the EBI Platform. Only Users explicitly defined within M/C are granted access. This enables us to insure data does not fall into the hands of users, who do not fully understand its intent. Therefore, while they may have access to the server on which the repository resides, they will not be permitted access to the SAS repository, without explicitly being granted permissions via M/C. Depending upon your needs, you may or may not choose this approach. However, you need to carefully consider the pros and cons of this strategy. We erred on the side of caution, choosing this more conservative and secure route.

We recommend establishing a naming convention for your user groups, to facilitate navigation within M/C. Establishing a naming convention which includes such things as user type or functional/business area will allow for common user groups to be “grouped” together, when sorted in M/C.

Let’s expand on our example.

You have decided to establish only one group for the Super Administrator user type. Only the primary and backup SAS Administrators will be members of this group.

You have established two groups for the Administrator user type. Both groups consist of IT associates supporting specialized business applications.

You have established a number of groups for the Super User type, one for each business area, which includes such areas as: Contracts, Cost Accounting, Finance, Manufacturing, Purchasing, Order Management and Training. In general, the members of these groups are in positions which call for them to have more broad access to information, and thereby, have been granted higher authority, as compared to End User groups.

For each Super User group, you have established a corresponding group for the End User Type. In general, the members of these groups are in positions which call for them to have less broad access to information, and thereby, have been granted a lesser authority, as compared to Super User groups.

For the administration of SAS Portal Content, which is discussed in more detail under the section “**PORTAL SECURITY**”, you have decided to create an individual generic identity for each of the business areas -- for example, “Portal Content Administrator for Contracts”. This identity is a member of *both* the Super User and End User groups for the business area, for example, the Contracts business area. This is an exception to the recommendation, to assign identities to *one and only one* group. You can read more about this under the section “**USERS**”.

You have also established an All Users group for each of the business areas. Both the Super User and End User groups are members of the business area’s All User group. Additionally, the Portal Content Administrator for the business area is also a member of this All Users group. You are establishing the concept of an “All” Users group to facilitate the implementation of Portal Content sharing.

Finally, you have established three high level groups: one whose members are the “All Users” groups for each of the business areas; one whose members are the “Super Users” groups for each of the business areas; one whose members are the “End Users” groups for each of the business areas.

The establishment of the above three groups, should drastically minimize the number of entries required in the Default Access Control Template, while at the same time, providing the flexibility required for the administration of security, across the board. For more details, see the section “**USE OF ACCESS CONTROL TEMPLATES**”.

USERS

For each user needing access to the EBI Platform, you need to establish an identity within M/C’s User Manager. When you initially create each identity, it is strongly recommended that the initial password adhere to your company’s security policies. Once you have established the identity in the repository, you will need to install SAS Personal Log-In Manager, on the user's PC. The users can then invoke SAS Personal Log-In Manager to change his/her password.

When you identify a user as needing access to the EBI Platform, you should ask yourself the following questions:

1. Which type of user is this person? – see "**USER TYPES**" for more information on available user types.
2. To which group should the user belong? -- see "**USER GROUPS**" for more information on available groups.

It is suggested that a user identity belong to one and only one group. This strategy will eliminate the risk of a given identity's authorization being conflicted as a member of multiple groups. One identity belonging to multiple groups has been found to cause unpredictable results. Exceptions to this recommendation should be thoroughly tested to insure results are as desired.

Aside from user identities established for production, we suggest you consider establishing user identities for training purposes.

In our example, you have decided to establish 20 user identities, all of which belong to the Training User Group. Depending upon the training needing to be conducted, the identities are assigned as members of either the Super Users Training Group or the End Users Training Group.

USE OF ACCESS CONTROL TEMPLATES (ACTs)

If you have not considered the use of Access Control Templates (ACTs) for the assignment of permissions, we highly recommend it. Such a strategy can minimize the number of entries, in the Default Access Control Template (ACT) (aka Repository ACT), while at the same time, providing the flexibility required for the administration of security across all metadata objects and software components. What follows is a discussion involving the use of the default ACT in conjunction with, other overriding ACTs, as a potential approach to security administration.

Permissions granted in the Default ACT will be inherited by all metadata objects throughout the metadata repository, unless they are explicitly denied or "re-granted", by an overriding ACT or an overriding ACE (Access Control Entry) at the object level. The "re-granting" of permissions may be required, if you use an overriding ACT to limit authority for a given group to "read-only" at a higher level parent folder structure, yet have the need for that group to gain "write" access to objects belonging to a lower level child folder structure belonging to the higher level parent. We will review an example of this in the section "**SECURITY RELATED TO STORED PROCESSES**".

Of importance to note, is if you have an individual or group requiring only Read and ReadMetadata for all but a few metadata objects, which in addition to Read and ReadMetadata also require Write and WriteMetadata permissions, that individual or group must be granted all four permissions in the Default ACT, in order for the individual or group to be granted Write and WriteMetadata for the *exception* metadata objects. You cannot grant permissions to any metadata object for an individual or group, unless that permission has been granted in the Default ACT. Therefore, you will need to: establish an overriding ACT to revoke all but Read and Readmetadata; apply that overriding ACT to all objects within the repository; establish an overriding ACT to "re-grant" Write and Writemetadata; and apply that overriding ACT to the *exception* metadata objects. This may seem counter-intuitive, but this is how it works.

With the exception of the individual identities included as part of the default ACT as delivered by SAS, we would recommend restricting all other default ACT entries on the "Users and Permissions" tab to groups. Avoid adding individual identities, as on-going maintenance efforts will increase significantly.

In our example, your Default ACT consists of the following entries:

1. All predefined SAS Users/Groups and associated recommended permissions
2. "GROUP – COMMUNITY – End Users" – Members of this group will only include the End User Groups and will inherit the permissions granted to this group. The permissions granted to this group are ReadMetadata, WriteMetadata, Read and Write.

3. “GROUP – COMMUNITY – Super Users” – Members of this group will only include the Super User Groups and will inherit the permissions granted to this group. The permissions granted to this group are ReadMetadata, WriteMetadata, Read and Write.
4. “GROUP – Administrators – IT” – Members of this group will only include the Administrator identities and will inherit the permissions granted to this group. The permissions granted to this group are ReadMetadata, Create, WriteMetadata, Read, Write, Delete.
5. “GROUP – Super Administrators – IT” – Members of this group will only include the Super Administrator identities and will inherit the permissions granted to this group. The permissions granted to this group are ReadMetadata, CheckInMetadata, Create, Administer, WriteMetadata, Read, Write, Delete.

Additionally, in our example, you will make use of overriding ACTs. Consider the establishment of overriding ACTs similar to the following, to facilitate the granting/denying of inherited permissions to/from metadata objects. It is highly recommended the names you assign to your overriding ACTs reflect exactly what they do, when applied to a particular metadata object.

1. Super Administrators Deny all but RM/R (ReadMetadata/Read)
2. Super Administrators Grant Default (meaning grant Default ACT permissions)
3. IT Administrators Deny
4. IT Administrators Deny all but RM/R
5. IT Administrators Grant Default
6. COMMUNITY – Super Users Deny
7. COMMUNITY – Super Users Deny all but RM/R
8. COMMUNITY – End Users Deny
9. COMMUNITY – End Users Deny all but RM/R

LIBRARIES

When establishing the metadata for your libraries, you have a variety of library engine types from which to choose, including SAS, Oracle, ODBC, Teradata, etc. The number of libraries you define will depend upon your shop’s needs. In addition to defining the metadata for the library, certain library engine types, such as Oracle and ODBC, require you to also define corresponding database schemas, while libraries of the Base SAS engine type do not require corresponding database schemas. Consult your documentation for more information regarding when you do and do not need to define database schemas.

We recommend the administration of security for all of your libraries be accomplished through the application of Overriding ACTs. As you may recall from the **“USE OF ACCESS CONTROL TEMPLATES (ACTs)”** section of this paper, by default, metadata objects such as libraries inherit permissions granted in the Default ACT. However, when it comes to libraries, chances are, you will not want all users to see the content of all libraries. Therefore, on the authorization tab of the library properties, we would suggest you apply Overriding ACTs to achieve the desired level of security, to appropriately control access to library content. Only in rare cases, should you consider incorporating the application of Access Controls Entries (ACEs) to further refine the permissions on libraries. The application of ACEs should be kept to a minimum, in the interest of keeping the administration of security as straightforward as possible.

When defining a library, you have the option of pre-assigning the library. Pre-assigning libraries will make their contents immediately available in such tools as SAS Enterprise Guide, without user intervention. Library pre-assignment can be achieved through the library’s metadata definition and autoexec files. A watch-out regarding the pre-assignment of libraries is that each time a user accesses an EBI reporting tool, such as SAS Enterprise Guide or the SAS Add-In for Microsoft Office, those pre-assigned libraries to which the user has access, will initiate a “session” in the corresponding database, such as Oracle, SQLServer or Microsoft Access. In certain cases, the initiation of sessions in the corresponding database could have adverse effects. Therefore, the decision to pre-assign a library must be made carefully and with a full understanding of the ramifications of pre-assignment.

Let’s continue with our example.

You have decided that Super and End Users are permitted to view selected Oracle libraries. They also have access to the library established to house content related specifically to their business/functional area. You chose to establish libraries for each business area, to keep library content appropriately segregated to facilitate the administration of security around sensitive data. For example, members of the End Users group for finance and Super Users group for finance would have access to the finance library, which would house content specific to the Finance Department. Members of other functional areas would not have access to data in the finance library.

SECURITY RELATED TO INFORMATION MAPS

The version of the EBI Platform that you are own, will drive your use of Information Maps. In earlier versions of the EBI Platform, Information Maps were only used in conjunction with Web Report Studio. In more recent versions of the EBI Platform, the accessibility to Information Maps has increased.

For our example, you have decided on a very straightforward, simplistic use of Information Maps. You will establish an Information Map folder corresponding to each library defined within the Data Library Manager. The security used to access Information Map folders, will be based upon that used to grant/deny access to/from the library corresponding to the Information Map folder.

SECURITY RELATED TO STORED PROCESSES

Metadata definitions related to Stored Processes can be found by traversing the BIP Tree within BI Manager, until reaching the StoredProcesses folder (BI Manager → BIP Tree → Report Studio → Shared → Reports → StoredProcesses).

We suggest the administration of security for your Stored Processes be accomplished through the application of Overriding ACTs. As you may recall from the **"USE OF ACCESS CONTROL TEMPLATES (ACTs)"** section of this paper, by default, metadata objects inherit permissions granted in the Default ACT. However, when it comes to Stored Processes, you may not want all users to see all Stored Processes. On the authorization tab of all folders in the "BI Manager → BIP Tree → Report Studio → Shared → Reports → StoredProcesses" path, you could apply appropriate Overriding ACTs, so all user groups, only have Read and ReadMetadata permissions. By setting permissions in this manner, it prohibits users from inadvertently saving Stored Processes content to an upper level folder.

We would also suggest the creation of Stored Processes be limited to those individuals within your organization who have solid programming skills. We would not recommend granting authority to develop Stored Processes to your general user community.

That said, once again, let's expand on our example.

Within the "StoredProcesses" folder, you have decided to establish one folder for each of your functional/business areas. Additionally, you will establish a few "special use" folders.

You will create one special use folder to be used to house Stored Processes being developed by members of your IT Department, as only members of your IT Department will be permitted to create/modify Stored Processes. Security has been established via the application of Overriding ACTs to limit access to this development folder to Super Administrators and Administrators.

Once the Stored Processes are ready to be tested by the user community, they will be moved to a Stored Process User Testing folder, for user acceptance testing. Security has been established via the application of Overriding ACTs to grant full access to Super Administrators and Administrators and Read/ReadMetadata to all Super and End Users, permitting them to conduct necessary testing.

Once the Stored Process is ready for production, it will be moved into one of the production folders, depending upon its intended use. If the Stored Process is to be executed, yet not rendered to the user community, it will be moved to one of two special use folders, intended to house such Stored Processes. Security for both of these special use folders is set to prohibit access from Super and End Users. If the Stored Process is to be available to all users, regardless of their functional area, it will be moved to the General Use Stored Process folder. Stored Processes in the General Use folder are available for execution by all users.

If a Stored Process was established for a specific functional area, it will be moved to that functional area's Stored Process folder, where content of that folder is viewable only by Super and End Users in that functional area.

You will achieve the desired level of security by coupling the application of Overriding ACTs on the authorization tab of each functional area's Stored Process folder, with the addition of the specific functional area Super and End User groups, to the names pane in the authorization tab, limiting the viewing of the folder to only Super Administrators, Administrators and members of the functional area's Super and End User groups.

SECURITY RELATED TO WEB REPORT STUDIO

Metadata definitions related to Web Report Studio (WRS) can be found by traversing the BIP Tree within BI Manager, until reaching the Reports folder (BI Manager → BIP Tree → Report Studio → Shared → Reports).

We suggest the administration of security for your WRS content be accomplished through the application of Overriding ACTs. As you may recall from the **"USE OF ACCESS CONTROL TEMPLATES (ACTs)"** section of this paper, by default, metadata objects inherit permissions granted in the Default ACT. However, when it comes to WRS content, you may not want all users to see all WRS content. On the authorization tab of all folders in the "BI Manager → BIP Tree → Report Studio → Shared → Reports" path, you could apply appropriate Overriding ACTs, so all user groups only have Read and ReadMetadata permissions. By setting permissions in this manner, it prohibits users from inadvertently saving WRS content to an upper level folder.

Let's continue with our example.

Within the "Reports" folder, you will establish one folder for each of your functional areas to house WRS content. Within each functional area's WRS folder, you will establish a WRS Super User folder to house WRS Reports to be used by that functional area's Super Users. Also, within each functional area's WRS folder, you will establish a WRS End User folder to house WRS Reports to be used by that functional area's End Users, as well as, that area's Super Users. Additionally, you will establish a folder intended for use by IT, to house WRS reports under development.

The IT development folder will be established for use by members of IT, when their services are required by the business community to develop a WRS report, which is more complicated than those usually developed by Super Users and End Users. This folder and its contents would only be seen by members of IT, and Overriding ACTs would be applied on the authorization tab of the folder, to enforce this strategy. Once WRS reports being developed by IT are ready to be moved to production, they would be saved to the appropriate WRS functional area folder.

For each functional area, a WRS folder structure would been established to promote the sharing of WRS content among users within a given functional area, without burdening users in other functional areas, with having to sift through content of no interest to them. Super Administrators, Administrators and Super Users for a given area would be able to create/change WRS reports and make them available to other users in their functional area. End Users could execute WRS reports located in their area's End User Folder. End Users could also create WRS reports for their own personal use. However, these reports would be located in a folder, specific to that user and would not be able to be shared with others, without intervention on the part of a Super Administrator or Administrator.

You will achieve the desired level of security by coupling the application of Overriding ACTs on the authorization tab of each functional area's WRS folder, with the addition of the specific functional area Super and End User groups to the names pane in the authorization tab, limiting the viewing of the folder to only Super Administrators, Administrators and members of the functional area's Super and End User groups. Permissions would also be set so that users able to see the folder's content, could only see content and would not be able to inadvertently save WRS content to this functional area's upper level folder.

The Super User Folder for the functional area and End User Folder for the functional area would inherit permissions from the parent level Folder for the functional area, which for all authorized users are Read and ReadMetadata. Since it will be at the Super and End User Folder levels that you will store WRS content, you will apply Overriding ACTs to "re-grant" Super Administrators and Administrators permissions they receive from the Default ACT, including Write and WriteMetadata.

Permissions would be explicitly set on the functional area's Super User Folder to allow the functional area's Super Users Write and WriteMetadata, enabling them to create and modify WRS reports, for use and viewing by other Super Users in their area. Permissions would also be set to deny access from the functional area's End Users.

Permissions would be set on the End User Folder to allow the functional area's Super Users Write and WriteMetadata, enabling them to create and modify WRS reports for use and viewing by other Super Users in their area. Permissions would also be set to grant Read and ReadMetadata to End Users belonging to the functional area, enabling the End Users to execute WRS Content in this folder, but to not create or change content.

PORTAL SECURITY

The establishment of a strategy for the sharing of content within the Portal, will probably prove to be one of your biggest challenges. We would highly suggest creating a process map of how you would like to implement the Portal at your organization and reviewing with your entire implementation team. You need to have a thorough understand of how the various aspects of security for the Portal relate to one another and what exactly you would like to accomplish with the roll out of the Portal to your user community.

To illustrate a potential implementation plan for the Portal at your organization, let's continue with our example.

In our example, you want to be able to:

1. Not limit who is able to share Portal Content within a given functional area
2. Share certain Portal content with only End Users in a given functional area
3. Share certain Portal content with only Super Users in a given functional area
4. Share certain Portal content with both End and Super Users in a given functional area
5. Share certain Portal content with all users, regardless of their functional area.

Because you do not want to limit who is able to share Portal Content within a given functional area, you need to create a generic user identity for each area, whose sole purpose would be to share Portal content with groups within the functional area and which could be used by appropriate users within the area. In order for one identity to share content with multiple groups, that identity needs to be a member of every group to which content is to be shared. Therefore, the Portal Content Administrators would be the exception to the "a user can belong to one and only one group" rule. This works because Portal Content Administrator identities have no power within the EBI platform, other than the sharing of portal content. The Portal Content identity for each functional area would be added as a member of the End User Group for the functional area, the Super User Group for the functional area and the ALL USERS Group for the functional area.

The identity would then be added to the authorization tab of the Permission Tree properties for the above three groups, with ReadMetadata and WriteMetadata granted as permissions. This last step would activate the identity as the Content Administrator for each group. Therefore, when content is to be shared with only End Users for a particular area, the Content Administrator for the functional area shares the content with the End User Group for the functional area. When content is to be shared with only Super Users for a particular area, the Content Administrator for the functional area shares the content with the Super User Group for the functional area. When content is to be shared with both End and Super Users for a particular area, the Content Administrator for the functional area shares the content with the ALL USERS Group for the functional area.

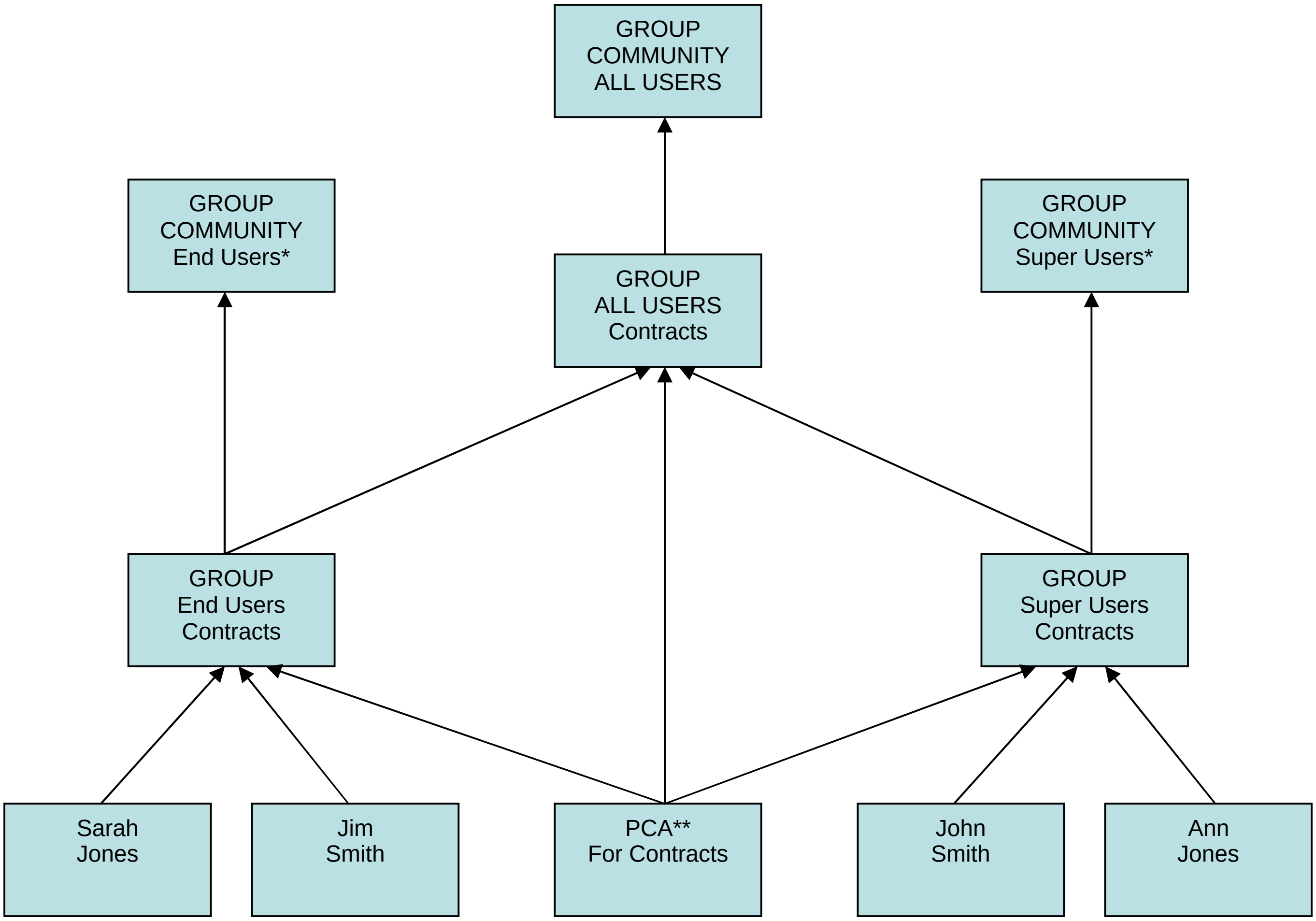
Finally, if you have a need to share Portal content with all users regardless of their functional area, you could utilize the Portal Administrator identity, SAS Web Administrator. The SAS Web Administrator shares content with the COMMUNITY ALL USERS Group and since all of the ALL USERS groups are members of COMMUNITY ALL USERS Group, the content is shared with ALL users.

To further illustrate our example, please review the hierarchical structure below. The example assumes only two functional areas:

1. **GROUP – COMMUNITY – ALL USERS** has the following members:
   a. GROUP – ALL USERS – Contracts
   b. GROUP – ALL USERS -- Manufacturing
2. **GROUP – COMMUNITY – Super Users***** has the following members:
   a. GROUP – Super Users – Contracts
   b. GROUP – Super Users – Manufacturing
3. **GROUP – COMMUNITY – End Users***** has the following members:
   a. GROUP – End Users – Contracts
   b. GROUP – End Users – Manufacturing
4. **GROUP – ALL USERS – Contracts** has the following members:
   a. GROUP – Super Users – Contracts
   b. GROUP – End Users – Contracts
   c. Portal Content Administrator for Contracts
5. **GROUP – ALL USERS – Manufacturing** has the following members:
   a. GROUP – Super Users – Manufacturing
   b. GROUP – End Users – Manufacturing
   c. Portal Content Administrator for Manufacturing
6. **GROUP – Super Users – Contracts** has the following members:
   a. John Smith
   b. Ann Jones
   c. Portal Content Administrator for Contracts
7. **GROUP – Super Users – Manufacturing** has the following members:
   a. Linda Jones
   b. Eric Smith
   c. Portal Content Administrator for Manufacturing
8. **GROUP – End Users – Contracts** has the following members:
   a. Sarah Jones
   b. Jim Smith
   c. Portal Content Administrator for Contracts
9. **GROUP – End Users – Manufacturing** has the following members:
   a. Debbie Smith
   b. Rick Jones
   c. Portal Content Administrator for Manufacturing

*******Listed as a “user” on the Default ACT.

For additional clarity regarding the hierarchical structure, refer to the diagram on page 10, which focuses on the Contracts functional area.



* Has entries in Default ACT
** Portal Content Administrator

CONCLUSION

SAS Management Console is a powerful tool and, if understood, can be used to tailor metadata object security to support your organization's specific needs. The importance of planning and the development of strategies for the multiple EBI components prior to setting up security for the metadata cannot be stressed enough.

CONTACT INFORMATION

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

Lisa Frank
BD
7 Loveton Circle
Sparks, MD 21152
Work Phone: (410) 316-4203
E-mail: Lisa_L_Frank@bd.com