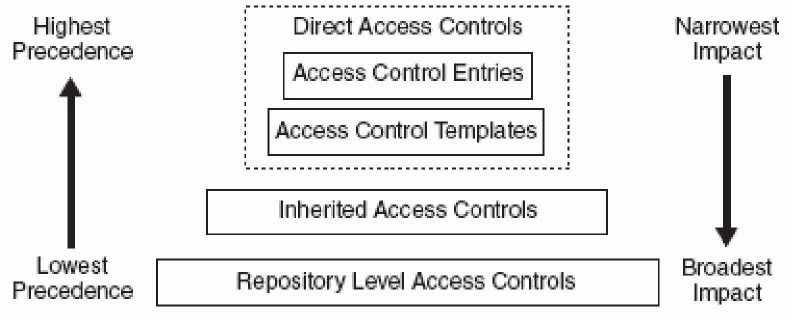
When designing the metadata architecture several years ago, the goal was to provide a flexible architecture that would not have to be tied to a hierarchical structure. To achieve this, object associations were used to tie different metadata elements together. Some associations provide additional context, and some associations provide security inheritance.

To determine authorizations, we first look for a direct access control entry (ACE) or access control template (ACT) on the object. If an ACE or an ACT does not exist, security associations are searched until an ACE is found. If all associations are traversed without finding explicit authorizations, then the authorizations in the DefaultACT will apply. The following diagram from the *SAS 9.1.3 Intelligence Platform: Security Administration Guide*[1] shows the order of precedence.

Figure 3.1 Access Controls in the Metadata Authorization Layer



If authorizations are not placed directly on an object, security inheritance is based on the object's security associations. Objects can be associated to a folder in a content tree and/or associated to another "parent" object. If you update an association that participates in security inheritance, you must have WriteMetadata (WM) authorization to both the original object and the associated object. For example, to register a library to a server context, you must have WM authorization granted on both the library and the server context.

Metadata Security Implications in SAS 9.1.3

Due to this architecture, you must grant ReadMetadata (RM) and WM authorizations in the repository-level access control template (DefaultACT) for any groups and users that will read and write metadata.

- There are multiple object types, such as access control templates, server contexts, users, and groups, that do not have security associations to a content tree. These "loose" objects have authorizations applied directly from the DefaultACT.

- When new objects are created, authorizations are checked when the object is first created, before associations are added. Therefore, by default, newly created objects have authorizations applied directly from the DefaultACT. This means that any user who creates a new object must have WM authorization granted at the repository level.

- Objects that you see in end-user applications (such as reports, stored processes, ETL jobs, and so on) are called "logical objects" because they are referenced as a single metadata object, but they are actually made up of multiple, individual metadata elements. Some of these elements (such as resource templates for defining libraries and servers) are also "loose" objects, such that their authorizations are determined only by the DefaultACT. This means that WM authorization must be granted in the DefaultACT to update these associations.

[1] http://support.sas.com/documentation/configuration/bisecag.pdf

- There are multiple factors that drive the need to grant WM authorization in the DefaultACT for Portal users.
  a. Portal users need the ability to create and manage profile objects.
  b. Each Portal user has a permission tree that is used to secure Portal content items. These trees are created and stored in the "Portal Application Tree" root folder.

  Because both these scenarios involve creating new objects (profiles and permission trees), WM authorization must be granted in the DefaultACT.

Best Practices for Setting Authorizations in SAS 9.1.3 for SAS Enterprise BI Server (or BI Server) Installations

In SAS 9.1.3, several of the SAS®9 BI applications do not support custom repositories, therefore, all content must be stored in a single repository (Foundation repository). In addition, separate applications can be configured to look for their objects at specific root locations in the content tree. This means that security might need to be mirrored across multiple branches of the content tree. To help minimize the administration of security privileges, here are some best practices to follow.

❖ **Take care of system administration accounts**
By default, a group called SAS System Services should exist in the DefaultACT with RM authorizations. Members of this group should be the SAS Trusted User and SAS Web Administrator. These accounts will read metadata on behalf of others, but they do not need WM authorization.

If you are using the Portal, you should include the SAS Guest User in the DefaultACT with RM authorization. This account is typically used to launch the public kiosk page and will need access to public content.

You also need to create a group for system administrators. These are users who will be maintaining the system, and they should be granted RM and WM authorizations in the DefaultACT.

❖ **Determine what authorizations you want to give to PUBLIC users**
PUBLIC users are all users who can authenticate to the metadata server. They do not have to have an identity explicitly registered in metadata. To determine what authorizations to set for PUBLIC users, follow these guidelines.

- If PUBLIC users will not be allowed to access SAS content, then you should deny RM and WM authorizations in the DefaultACT.
- If PUBLIC users will only be allowed to view SAS content, then you should grant RM and deny WM authorizations in the DefaultACT.
- If PUBLIC users will be allowed to create, modify, and view SAS content, then you should grant RM and WM authorizations in the DefaultACT. You can still restrict access to content at a lower level in the repository.

❖ **Determine what authorizations you want to give to the SASUSERS group**
SASUSERS are users who have identities registered specifically in metadata. To determine what authorizations to set for SASUSERS, follow these guidelines.

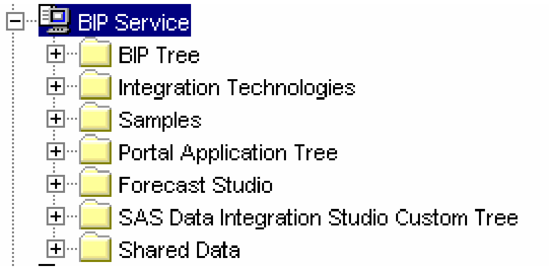
- If SASUSERS will not be allowed to access SAS content, you should deny both RM and WM authorizations in the DefaultACT. Then, you should specify which groups (or users) can access content, and add them explicitly to the DefaultACT with the appropriate authorizations.
- If SASUSERS will only be allowed to view SAS content, then you should grant RM and deny WM authorizations in the DefaultACT.
- If SASUSERS will be allowed to create, modify, and view SAS content, you should grant both RM and WM authorizations in the DefaultACT. You can still restrict access to content at a lower level in the repository.

- **Apply security measures to groups so that only administrators can modify membership**
  Security measures should be applied to restrict general users from being able to modify group membership. If you have already set up groups in metadata, you can use the %MDUGRPAC macro to programmatically apply permissions to your identity groups. This macro can be found in the SAS install directory. Here is an example path on Windows: \Program Files\SAS\SAS 9.1\core\sasmacro\

  For more information about the %MDUGRPAC macro, see the documentation at : http://support.sas.com/onlinedoc/913/getDoc/en/bisecag.hlp/a003094012.htm

- **Applying authorizations in the next level down from the DefaultACT**
  Because you will probably need to grant RM and WM authorizations in the DefaultACT for your users, let's look at the next level where you can start denying access on a broad basis. Here are some places to consider:

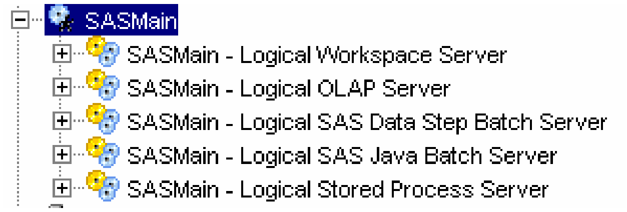
  - BIP Service software component
    This object represents the root of several content folders that have both BI and ETL content. This includes SAS Web Report Studio reports, information maps, stored processes, data explorations, ETL jobs, tables, and user transformations.



BIP Service
- BIP Tree
- Integration Technologies
- Samples
- Portal Application Tree
- Forecast Studio
- SAS Data Integration Studio Custom Tree
- Shared Data

  - ETL Studio software component
    This object represents the root of the SAS Data Integration Studio Custom Tree, and participates in inheritance for ETL jobs, tables, and user transformations.

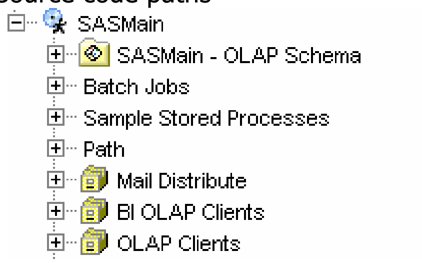
  - SASMain Server context
    The SASMain server context participates in inheritance for a number of different objects that do not participate in content tree inheritance. These include the following objects:
    - Servers



SASMain
- SASMain - Logical Workspace Server
- SASMain - Logical OLAP Server
- SASMain - Logical SAS Data Step Batch Server
- SASMain - Logical SAS Java Batch Server
- SASMain - Logical Stored Process Server

    - Libraries and Schemas

    - Source code paths



SASMain
- SASMain - OLAP Schema
- Batch Jobs
- Sample Stored Processes
- Path
- Mail Distribute
- BI OLAP Clients
- OLAP Clients

Other considerations

- In order to navigate to an object, you must have RM authorization for the entire path. For example, if you want to navigate to a report for SAS Web Report Studio, you would need to have RM authorization to BIP Service and the folders BIPTree, ReportStudio, Shared, and Reports. If any one of these locations is not accessible, then the path is broken and the report cannot be opened.
- To create an object in a folder (by creating a new object or moving an existing object), you need to have WM authorization on the target folder.
- Several object types can inherit authorizations from servers. Therefore, you must have WM authorization to the server if you want to create new objects of these types. You will not be able to create new library or stored process definitions if you do not have WM authorization to the servers.
- For SAS Web Report Studio, you can configure the default location for information maps to be in the same location as your reports. This will enable you to maintain a single reporting content tree to use for controlling access. There are two possible configurations:
  - Information Maps: *ReportStudio/Shared*; Reports: *ReportStudio/Shared*
    This requires that all information maps be stored in a shared area. You cannot save information maps to your private user space and have them available to SAS Web Report Studio.
  - Information Maps: *ReportStudio*; Reports: *ReportStudio/Shared*
    This allows you to save information maps to any report location (shared or private). However, because this does surface the user areas for SAS Web Report Studio, you need to make sure these have the appropriate security measures applied.