

Securing SAS®9 Business Intelligence Content Managed in Metadata

Information in this document pertains to SAS 9.1.3 Service Pack 4.

SAS®9 client software uses metadata to manage and secure business intelligence (BI) content items. Secure access to content items through client user interfaces is controlled by metadata permissions, which are enforced by the SAS Metadata Server. ReadMetadata and WriteMetadata access controls that are placed on individual items or folders grant or deny access to content items based on the identity of the user. Content items managed in metadata include:

- Stored Processes
- Information Maps
- Data Explorations
- Web Reports
- Publication Channels and archived Result Packages

Data tables and libraries are not discussed in this document, but information about them is given in SAS Data Integration Studio (formerly named SAS ETL Studio) documentation. For information about controlling OLAP Cube access in metadata, see the *SAS Intelligence Platform: Security Administration Guide* available at support.sas.com/documentation/configuration/bisecag.pdf.

Usage Patterns

Client applications create objects in the metadata repository to model and provide authorization control to content items. Some applications have specific repository locations for content metadata; other applications let the SAS administrator create and manage the locations.

SAS administrators must manage groups of users. Each group requires secure access to specific sets of content. The administrator creates sub-folders in the metadata root folder and applies access permissions so that PUBLIC is denied access while specific group access is granted. The administrator should also grant access to an Administrators group for content management. For more information about applying access controls, see the online help for the Authorization Manager plug-in to SAS Management Console.

Access to content is controlled by two permissions: ReadMetadata and WriteMetadata. These permissions are enforced by the server based on the identity of the connecting client.

- ReadMetadata permission is needed to navigate and read content. If a user does not have ReadMetadata for a content item, the item is not found in a search and is not viewable in a metadata browse client.
- WriteMetadata permission is needed to create new objects such as trees (folders) or content objects, such as Stored Processes, Reports, and Information Maps.

A first step in providing secure access to content is to secure who can modify group identities and membership. Group identities should be secured so that only administrators can modify membership.

To restrict group identity editing to administrators, complete the following steps:

1. Start SAS Management Console.
2. In the User Manager plug-in, select the group and then select Properties.
3. On the Authorization tab, select Grant WriteMetadata to Administrator groups, then select Deny WriteMetadata to PUBLIC.

Apply a similar security pattern to Access Control Templates (ACTs) that are applied to objects and folders. You can access ACTs by using the Authorization Manager plug-in to SAS Management Console.

1. For each ACT, select the Properties window and the Authorization tab.
2. Select Grant WriteMetadata for Administrator groups.
3. Select Deny WriteMetadata for the PUBLIC group.

Some SAS applications have a specific location for shared group content; other applications allow the Administrator to set the location. Some applications provide a User folder to store personal content items that are created in specific locations in the repository as presented in the section “Securing Content Locations in the Metadata Repository”. This document reviews these locations and provides strategies for securing content at these locations.

Access Controls and the Repository Default ACT

You can use Access Control Templates (ACTs) to apply sets of access controls to objects. The Default ACT has a unique role in that it applies (through inheritance) to all objects in the repository. Whenever a new object is created, the access controls of the Default ACT govern the initial creation of the object. This means that the identity that creates objects in the repository must be granted WriteMetadata in the Default ACT.

Granting WriteMetadata in the Default ACT

Customer administrators often ask why broad ReadMetadata and WriteMetadata permissions are needed in the Default ACT for operation of client software.

- ReadMetadata is required in order to navigate and search for objects.
- WriteMetadata is required for the initial creation of objects in the repository.

Parent objects like Servers and Folders provide permissions through inheritance down to child objects. After an object is created and associated with a folder, permissions are inherited from the folder as well as from the Default ACT.

For example, consider the creation of a Stored Process object. SAS Enterprise Guide users must have WriteMetadata in the Default ACT to create the object. To associate that object with a folder, users must have WriteMetadata for these objects also. After a user associates the object with a folder, that object inherits permissions from the parent object as well as the Default ACT.

The following table describes when and why identities must have WriteMetadata in the Default ACT.

Client Task

Requirement for WriteMetadata in Default ACT

Creating Stored Processes

Author of Stored Process needs WriteMetadata for initial creation of the Stored Process object.

Information Delivery Portal Personal Desktop

Portal users manage personal desktop configuration in the Portal profile object that is associated with each portal user's identity. WriteMetadata is needed in the Default ACT for creating and editing the profile object.

Information Delivery Portal Permission Trees

The 'Portal Application Tree' folder is accessed by Portal code to create user and group permission tree folders that are used to secure Portal content on a per-user and per-group basis. Portal users must be granted WriteMetadata permission for the 'Portal Application Tree' root folder when they first log on to the Portal. This is required because the user's permission tree folder is created as part of their first log-on process. A Portal user can be granted WriteMetadata in either of two ways:

- Directly granted by Access Control Entries (ACE) or ACT on a per-user or per-group basis
- Inherited from the Default ACT

Note: Because all Portal users must have a metadata identity (to save desktop personalizations), you can use the SASUSERS group (users with identities) to grant this required access permission. After a user's first log on, you can remove the WriteMetadata grant that allowed initial creation of the user's permission trees.

CAUTION: Be careful when changing the permissions for the 'Portal Application Tree'. The SAS Web Administrator identity and all Portal users (including SAS Guest) must have WriteMetadata on this root folder.

A best practice for applying access controls is to use folders (tree objects) to apply individual ACEs and ACTs to objects in the folders. Objects within a folder inherit access controls from the folder. For more information about access controls, inheritance of access controls, and precedence in inheritance, see the SAS Management Console Authorization Manager online help.

Most administrators will also apply access controls at the root of a folder path to provide additional and specific access controls to those objects within that folder that will augment those inherited from the Default ACT.

Note the two roles of the Default ACT:

- when creating an object, only the access controls of the Default ACT apply
- after creating the object and associating it with a folder or other objects, access controls are inherited from the Default ACT and from parent objects (folders, servers) through association

Setting Permissions for PUBLIC and SASUSERS Implicit Groups in the Repository Default ACT

The SAS Metadata Server supports two implicit membership groups:

- PUBLIC – All authenticated users
- SASUSERS – a subset of PUBLIC, authenticated users that have an identity in the repository

Administrators can choose to control access to content based on these groups. If ReadMetadata permission is denied for PUBLIC users but granted to SASUSERS, PUBLIC users will be able to log on to SAS Applications but will not be able to view any content. In this configuration, a user identity must be added to the repository in order for the application user to access content. This permission pattern (Deny PUBLIC ReadMetadata, WriteMetadata) can be applied through the Default ACT.

Securing Permission Patterns in Default ACT

The *SAS 9.1.3 Intelligence Platform: Security Administration Guide* (support.sas.com/documentation/configuration/bisecag.pdf) provides guidance for securing your metadata repository with permissions applied in the Default ACT. After completing these steps, the permissions in the Default ACT will be as follows:

- PUBLIC – Deny ReadMetadata, WriteMetadata
- SAS Guest – Grant ReadMetadata, WriteMetadata
- SAS Demo User – Grant ReadMetadata, WriteMetadata
- Administrators group (sasadm) – Grant ReadMetadata, WriteMetadata
- SAS System Services group (SAS Web Administrator, SAS Trusted User) – Grant ReadMetadata, WriteMetadata

With this configuration, the Demo User can log on to SAS Web Applications, and the SAS Guest account can be used to customize the Portal's Public Kiosk. The Web Administrator account can be used to manage Web Application content and be used as a utility account for managing security.

Note: The *SAS 9.1.3 Intelligence Platform: Security Administration Guide* includes content formerly in the *SAS 9.1.3 Enterprise Intelligence Platform: Administration Guide* before November 2006.

Two Approaches to Defining Group Access Controls

It is a best practice to grant permissions through groups. Using this approach, you can modify the Default ACT as follows:

Option 1: Identity-based groups

- PUBLIC – Deny ReadMetadata, WriteMetadata
- SASUSERS – Grant ReadMetadata, WriteMetadata
- Administrators – Grant ReadMetadata, WriteMetadata
- Portal Admins (SAS Web Administrator, customer Portal admins) – Grant ReadMetadata, WriteMetadata
- SAS System Services (SAS Web Administrator, SAS Trusted User) – Grant ReadMetadata, WriteMetadata

With this configuration, users who have identities in metadata (SASUSERS) have permissions to access the Portal and other Web Applications.

Administrators, including the Portal Admins group (SAS Web Administrator, SAS Trusted User), have permissions to manage Web Application security and operations. All Web Application users must have an identity in the SAS Metadata Server because PUBLIC users have been denied permissions. Most content folders will be accessible and we will describe permissions for securing content later in this document.

Option 2: Application user groups

PUBLIC – Deny ReadMetadata, WriteMetadata
SASUSERS – Deny ReadMetadata, WriteMetadata
Create “Web Application Users” group – Grant ReadMetadata, WriteMetadata. Add SAS Guest, SAS Demo User and all Portal and Web Application users.
Administrators group – Grant ReadMetadata, WriteMetadata
Portal Admins – Grant ReadMetadata, WriteMetadata
SAS System Services – Grant ReadMetadata, WriteMetadata

In Option 2, you have the same user permission pattern as in Option 1, but now you are creating specific application user groups (for example, Web Application Users), and granting permissions to the group instead of the broader permission grant to SASUSERS.

Note: You will need to create a group for each application user group in your system and grant them ReadMetadata at a minimum and WriteMetadata for certain activities.

Basic Permission Patterns for Securing Content Modeled in the Metadata Repository

ReadMetadata permission is required for navigating folder paths or searching for content items. For this reason, ReadMetadata permission is granted at root folders and then denied as needed to secure lower content folders.

It is a best practice to deny WriteMetadata permission to PUBLIC at the root folder level, and then grant it as required in lower folders. Users who only view content items do not need WriteMetadata permission. WriteMetadata permission is required to create, move, or delete a content object, and is therefore typically granted to content authors and administrators.

Use Access Control Templates (ACTs) to implement this best practice.

To use ACTs to restrict access at the root folder level, complete the following steps:

1. Start SAS Management Console.
2. In the Authorization Manager plug-in, create an ACT named “Public RM Only”.
3. On the Authorization tab, select Grant WriteMetadata to Administrator groups, then select Deny WriteMetadata to PUBLIC.

Apply this ACT to the root of content folders such as

/BIP Tree

The permission pattern of the “Public RM Only” ACT will be inherited from the root to subfolders and content within those subfolders. This prevents WriteMetadata access. The administrator then grants WriteMetadata to specific groups that need it. This permission pattern allows folder path navigation but blocks users from creating content in locations that are not secured or appropriate.

In lower folder locations, create ACTs for group folders by using the following group permission pattern:

- PUBLIC – Deny ReadMetadata, Deny WriteMetadata
- Group – Grant ReadMetadata– Can view but not alter
- Group administrators and content authors – Grant ReadMetadata, Grant WriteMetadata – Can create, delete, and copy

Using this permission pattern, application users can only navigate or search folders where they have ReadMetadata access. When navigating or searching, they will see in the user interface only the folders and content items that are available to their group.

The “group owner” permission pattern demonstrates three best practice rules for permission-based security:

Rule 1: Deny permission broadly (PUBLIC) and grant permission specifically (Group).

Rule 2: Use groups to control permissions and not individual identities. This is more manageable and easier to administer; users are added or removed from groups based on their need for role-based security.

Rule 3: Apply access controls to folders either directly or through inheritance, and let content items in the folder inherit access controls from the containing folder.

Flat and Hierarchic Group Folder Structure

There are two common patterns for applying access controls to group folders: flat folder structure and hierarchic folder structure.

A flat folder structure is most common and is easy to visualize and manage.

```
/BIP Tree/Groups/Dept A
/BIP Tree/Groups/Dept B
/BIP Tree/Groups/Marketing Division
```

A hierarchic folder structure typically follows the organizational structure for a business unit. There is a need to access content based on a hierarchy of role-based permissions. For example, work groups can only access group content, managers can access content across all groups plus a manager’s area, and executives can look at content across all group and managers’ areas plus an executive reports area.

```
/BIP Tree/ReportStudio/Shared/
  Reports/Sales
  Reports/Sales/National
  Reports/Sales/Southeast/Region
  Reports/Sales/Southeast/Florida
  Reports/Sales/Southeast/Georgia
```

These two approaches to folder structure each require a unique pattern of access permissions to properly secure the content folders. Examples that present these access permission patterns are provided in the Appendices.

Securing Content Locations in the Metadata Repository

SAS client applications can have default locations for storing metadata content entries, or the application administrator might be allowed to define these locations. Multiple SAS clients for each content type exist, and each client is designed to meet the needs of a specific user group. Some of these SAS clients are read-only viewers; other clients enable the user to create and edit content.

To understand, secure, and audit access to content, the application administrator must know how and where client applications access content objects and folders. The following section presents a summary organized by content type of SAS client software that views or creates content managed in metadata, and the typical locations of that content in the metadata repository. Use this summary to review these content locations and apply the appropriate access control patterns to secure the content.

Information Maps

Clients that Create Information Maps: SAS Information Map Studio, Data Explorer

Clients that View Information Maps: SAS Web Report Studio, SAS Information Delivery Portal, SAS Information Maps Navigator portlet and Tree Navigator portlet, Portal Search feature (Portal uses Data Explorer to view Maps surfaced in Navigator portlets and Search result lists); SAS Web OLAP Viewer for Java views OLAP maps.

SAS Web Report Studio Information Maps

The default repository location for maps used by SAS Web Report Studio is

```
/BIP Tree/ReportStudio/Maps/
```

Secure group subfolders can be created at this location. SAS Web Report Studio will search all subfolders at this location when providing the user with a list of data sources. A typical group folder pattern would be:

```
/BIP Tree/ReportStudio/Maps/Public  
/BIP Tree/ReportStudio/Maps/DeptA  
/BIP Tree/ReportStudio/Maps/DeptB  
/BIP Tree/ReportStudio/Maps/MarketingDivision
```

Data Explorer Information Maps

Maps used primarily by the Data Explorer can be stored in any location in the Information Service. Creating a Maps folder higher in a BI root folder makes navigation and access control easier.

```
/BIP Tree/Groups/DeptA – secured to Department A group (see group owner pattern in Appendix)  
/BIP Tree/Groups/DeptB – secured to Department B group
```

In this example, the location

```
/BIP Tree/Groups
```

can be secured so that only the administrator can create group folders and security permissions can be applied when group folders are created. A permission pattern to secure folder creation only for Administrators would be

- PUBLIC – Grant ReadMetadata, Deny WriteMetadata
- Administrators groups – Grant ReadMetadata, Grant WriteMetadata

Data Explorations

Visual Data Explorer and SAS Web OLAP Viewer for Java provide a File Save feature to store data explorations (specific views on information maps) in user folders created in the repository as

/BIP Tree/Users/<userid>/

For SAS 9.1.3 Service Pack 3, these folders are not secured to the owner. Apply the following permission pattern using an ACE to each folder after creation.

- PUBLIC – Deny ReadMetadata, Deny WriteMetadata
- User – Grant ReadMetadata, Grant WriteMetadata

SAS 9.1.3 Service Pack 4 includes enhancements for Visual Data Explorer (VDE) and SAS Web OLAP Viewer for Java limits the Save feature to a user's secure folder at /BIP Tree/Users/<userid>.

Both Data Explorer and SAS Web OLAP Viewer for Java provide a File Open feature. Because File Open access is controlled by ReadMetadata permissions, access controls should be placed on folders and content items such that the File Open feature provides secure access to content. Specifically, all users will have sole access to their own user folders and can also access group folders based on group membership.

Viewing and Saving OLAP Cubes

When Data Explorer or SAS Web OLAP Viewer for Java applications view OLAP cubes, information maps are generated for use by the query subsystem. These maps are created in the shared area

/BIP Tree/SASGeneratedMaps/<OLAP Schema Name>/

All users of Data Explorer and SAS Web OLAP Viewer for Java must have ReadMetadata, WriteMetadata access to this area when creating cubes. As a result, information maps that are created here are accessible to all users. Although the information map cannot expose cube attributes restricted by cube metadata permissions, application developers might want to limit access to these maps and limit this type of direct access to cubes.

Note: SAS 9.1.3 Service Pack 4 allows limiting direct access to cubes through an application parameter setting. In this configuration, the File Open dialog does not allow cubes in the Open option, only OLAP information maps.

Web Reports

Clients that Create Web Reports: SAS Web Report Studio

Clients that View Web Reports: SAS Information Delivery Portal Web Report Navigator portlet and Tree Navigator portlet

Note: The Portal Search feature also accesses Web Reports. By default, the Portal uses SAS Web Report Viewer to view web reports displayed in Navigator portlets and Search feature result lists.

The default location in the repository for shared Web Reports is,

/BIP Tree/ReportStudio/Shared/Reports/.

Secure group subfolders should be created at this location. For example:

/BIP Tree/ReportStudio/Shared/Reports/DeptA
/BIP Tree/ReportStudio/Shared/Reports/DeptB...

A Public folder can also be created with ReadMetadata and WriteMetadata granted to PUBLIC:

/BIP Tree/ReportStudio/Shared/Reports/Public

Such a folder would provide open access to any authenticated user to create and share content with all users. Other folders can be created and secured to share content within a group of groups.

When a user logs in for the first time, SAS Web Report Studio programmatically creates a personal user folder for that user. In the default folder naming structure, the location is

/BIP Tree/ReportStudio/Users/<userid>/Reports/

For SAS 9.1.3 Service Pack 3, the /Users folder has PUBLIC access by default, with each user's /Reports folder protected at creation by the following ACE settings:

- PUBLIC – Deny ReadMetadata, Deny WriteMetadata
- User – Grant ReadMetadata, Grant WriteMetadata

This protects the /Reports folder from view and access of content, but it does not secure the User folder name /BIP Tree/ReportStudio/Users/<userid> from view by another metadata browsing client such as the Information Delivery Portal's Web Report Navigator portlet. SAS 9.1.3 Service Pack 4 applies the user secured permission pattern one level up from the /Reports folder, which protects the user folder name from view.

Stored Processes

Clients that Create Stored Processes: BI Manager plug-in version 1.4 for SAS Management Console Stored Process Manager, SAS Enterprise Guide 3

Note: The BI Manager plug-in version 1.4 for SAS Management Console includes functionality previously contained in the Stored Process Manager.

Clients that View Stored Processes: SAS Add-In for Microsoft Office, SAS Information Delivery Portal Tree Navigator portlet and SAS Information Delivery Portal Stored Process Navigator portlet, and the SAS Information Delivery Portal Search feature. SAS Stored Process Web Application is used to view stored processes that are selected in Navigator portlets and Portal Search results lists.

Stored Processes can be created and accessed through any BI Root folder in the repository, requiring ReadMetadata to execute, and requiring WriteMetadata to create. The default location in the BIP Tree for stored processes used by Web Reports is:

/BIP Tree/ReportStudio/Shared/Reports/StoredProcesses/

You can create group folders in this location to organize and secure Stored Processes available for Web Reports.

Other BI root folders that are available in a typical SAS Enterprise BI Server install are:

- /Samples/Stored Processes/
Used by the SAS Integration Technologies install to store sample stored processes. Most customers will want to deny ReadMetadata to PUBLIC for the /Samples root, and only grant ReadMetadata to authoring groups (SAS Enterprise Guide users) that need to access the samples for example code.

- /Integration Technologies
SAS Publish-Subscribe metadata is stored in this root folder and used to manage Channels and Subscriber profiles. Deny WriteMetadata to PUBLIC in this folder area, and limit access to users who work with the publish framework.
- /Portal Application Tree
SAS Information Delivery Portal uses the root folder to maintain permission trees to secure Portal content. Deny WriteMetadata to PUBLIC for this folder and grant WriteMetadata to SAS Web Administrator, the Portal's utility administrator account. When permission tree folders are created, they are programmatically secured to the owning user or group.

CAUTION: Although it is possible, **DO NOT** store Stored Process objects in these repository folders. Best practice is to manage all content in the BIP Tree root folder. This provides a simpler environment to secure. Group folders can be maintained as

/BIP Tree/Groups/<group>

At this location, you can create group subfolders and apply permissions to

- Group users – Grant ReadMetadata
- Group administrators and content authors – Grant ReadMetadata, Grant WriteMetadata
- PUBLIC – Deny ReadMetadata, Deny WriteMetadata

As seen previously, this “group owner” permission pattern can be easily saved and applied as an ACT for each group.

/BIP Tree/GroupA – Apply GroupA Owner ACT

...

/BIP Tree/GroupZ – Apply GroupZ Owner ACT

Stored processes are unique in that they must have an associated Source directory object for the location of stored process source code. As a result, clients that create stored process objects must also create Stored Process Source directory objects. The directory objects obtain access controls only from the Default ACT, so users who are creating Stored Processes (SAS Enterprise Guide users, BI Manager plug-in for SAS Management Console users) must have WriteMetadata granted in the Default ACT.

Stored process objects are also unique in that they have an association to a Server object for execution (either SAS Workspace Server or SAS Stored Process Server). A user who is creating a Stored Process object must also have WriteMetadata granted for the Logical Stored Process Server for the execution server.

Finally, a stored process author must have WriteMetadata for the folder where the stored process is saved.

As a pattern of access controls, Stored Process authors (using the BI Manager plug-in for SAS Management Console or SAS Enterprise Guide) must be granted ReadMetadata and WriteMetadata for the following locations and objects:

- the Folder that contains the Stored Process entry
- the Logical Stored Process Server for execution
- the Default ACT for creating the associated source directory object

Creating and Securing Stored Processes with SAS Enterprise Guide

SAS Enterprise Guide 3 can be configured to use the SAS Metadata Repository to locate Workspace Servers and Stored Process Servers, and to save and share Stored Processes. Stored Process authors must first use the SAS Enterprise Guide Administrator to configure the SAS Metadata Repository as a project repository. The credentials that are provided for the Metadata Server connection should be those of the user and not a general purpose access account. This allows the use of metadata access controls to restrict Stored Process authors to group folders specific to their scope of work.

When a SAS Enterprise Guide Project is opened, a connection is made to the Metadata Server using the credentials that were defined by the SAS Enterprise Guide Administrator application. First-time users are prompted for these credentials, and these credentials are persisted by the application via default security settings.

The process of creating, testing, and delivering a Stored Process requires connections to both Workspace Servers (stores the SAS Enterprise Guide Project and executes the SAS code) and Stored Process Servers (executes the stored process). In SAS Enterprise Guide 3.01, users were prompted for server access credentials for the Workspace Server. For SAS Enterprise Guide 3.02, a connection to the Workspace Server is attempted with the cached Metadata Server credential, and then using available metadata logins for the user. When connecting to a Stored Process Server, cached credentials are not used but metadata logins will be used, and if none are available, the user is then prompted. Consistent credential caching and login management is in SAS Enterprise Guide 4.

The use of secured group folders restricts Stored Process authors to folders based on their scope of work, and allows secure delivery to the group. However, due to the nature of stored process objects and associations (to Servers and Source directories), Stored Process authors must have WriteMetadata for both the Stored Process folder and the Logical Stored Process Server where the Stored Process will execute and WriteMetadata in the Default ACT.

For SAS Enterprise Guide 3.0, a user must be granted WriteMetadata permission to all stored process entries that exist in a folder in order to save a new stored process in the folder. Because of this restriction, access controls for stored process entries should always be made at the folder level and passed by inheritance to each stored process entry in the folder. This ensures that consistent WriteMetadata access is provided to authors who create and save stored processes in the folder.

Publishing Framework Metadata

Clients that Create Publishing Framework: SAS Management Console Publishing Framework plug-in

Clients that View Publishing Framework: SAS Enterprise Guide 3, SAS Information Delivery Portal

Channels are managed in the SAS Integration Technologies BI root folder:

```
/Integration Technologies/Publish-Subscribe/Channels
```

Administrators can create channel entries using the SAS Management Console Publishing Framework plug-in. Administrators can create secure group sub-folders to organize and secure channel access for user groups.

Example:

```
/Integration Technologies/Publish-Subscribe/Channels/Sales  
/Integration Technologies/Publish-Subscribe/Channels/DeptA/WeeklyReports  
/Integration Technologies/Publish-Subscribe/Channels/DeptB/FinanceReports
```

The Publish-Subscribe model requires the following access control patterns:

- Subscriber or Admin must be granted WriteMetadata on the Package Subscribers folder in the Publishing Framework permission tree to create subscriber profiles
- Publishers must be granted ReadMetadata on the channel folder to read subscriber profiles for delivery
- Publishers to Channels with Archives (the content of which is tracked in Metadata) must be granted WriteMetadata on the Channel object

In the Portal, the Option “Manage Subscriptions” presents a list of Channels that users can subscribe to. The list is created from a search of channel objects in folders and sub-folders starting at the location:

/Integration Technologies/Publish-Subscribe/Channels

A user must be granted ReadMetadata permission to view the channel in the user interface, and must be granted WriteMetadata permission to subscribe to the channel.

Note: For this reason, ReadMetadata and WriteMetadata should always be granted or denied together for a channel object or a folder that contains channel objects that the administrator wants to offer for open subscription to a group. This provides consistency for the user interface that enables the user to subscribe to any Channel that is displayed to them.

Closed Enrollment Channels

Some channels might be offered with closed enrollment. A Portal administrator might want to manage a channel to which all Portal users are subscribed, or create a channel for news that is displayed on the Portal’s public kiosk. For these channels, the administrator would create a folder such as the following:

/Integration Technologies/Publish-Subscribe/Channels/AdminControlled

and place an ACT on this folder with the following permissions:

- Administrators groups – Grant ReadMetadata, Grant WriteMetadata
- PUBLIC – Deny ReadMetadata, Deny WriteMetadata

The administrator would then create two channels:

PortalNews – subscribe all portal users

PublicNews – only subscribe SAS Guest (account used for the public kiosk)

Only the administrator can change subscriptions for these channels. If users other than the administrator group need to publish to these channels, those users must be granted WriteMetadata permission.

The Portal “Manage Subscriber Profiles” feature enables a user to create subscriber profiles that provide information to publishing processes for content delivery. Alternatively, an administrator can create profiles for subscribers by using the Publishing Framework plug-in to SAS Management Console.

By default, the Subscriber folder is created without access controls. However, this enables users of clients with navigation user interfaces (such as VDE and Portal Tree Navigator portlet) to view subscriber profile names in this location:

/Integration Technologies/Publish-Subscribe/Subscribers/Content Subscribers

Note: If the administrator chooses to keep this folder location open by granting ReadMetadata permission, then subscriber profile names should not be based on full user name, userid, or other information that could expose user identity.

Publish-Subscribe Usage

Publishing from the Portal – Collaboration

In this low-security scenario, a group of users share a channel among themselves that is secure to the group, and all subscribers can also be publishers. Publishing from the Portal requires an archive because the Portal's Publication Channel Subscriptions portlet is used to view channel content. Create a folder and apply access controls that grant the group and administrators ReadMetadata, WriteMetadata while denying PUBLIC ReadMetadata, WriteMetadata. Within this folder, a channel is created, and a subscriber group is created for channel subscribers.

Alternatively, the Channel could be directly secured to the group (grant ReadMetadata, WriteMetadata for the group, deny ReadMetadata, WriteMetadata PUBLIC, grant Admins ReadMetadata, WriteMetadata). The group is granted ReadMetadata to the subscriber profile for each member of the group, while PUBLIC is denied ReadMetadata to the group's subscriber profiles.

As a low-security scenario, it does not matter if the subscriber information for group members is exposed to the group. If a DAV directory is used as the archive for the channel, it should be created and secured to limit access to group members and administrators.

If a Portal user's subscriber profile specifies e-mail as the publish delivery mechanism, the e-mail will contain a link to the Portal content. When the user first attempts to access the link, the Portal uses the SAS Guest account (used to manage the Portal's Public Kiosk page) to access the content. If this fails, the user is prompted to log on.

Using the SAS Guest account enables publishers to conveniently publish low-security content that is available to all users. If the channel archive is a DAV directory, then access must be provided to the SAS Guest account, and all publishers must know that any user who searches for content at the public kiosk will have access to the published package.

Content that must be secure to a specific group membership should be placed in a metadata folder that denies PUBLIC ReadMetadata and grants group members ReadMetadata. Further, SAS Guest must be denied ReadMetadata. Group users who receive content via URLs in e-mail will be prompted to log on for access, and the content will be secure from search from the public kiosk.

Publishing Channels for High Security and Assured Delivery

In some publishing environments, Channel content is sensitive and subscribers have limited access to channels. The same access control requirements might exist when there is a channel for important alert information and subscribers are not allowed to unsubscribe from the channel.

Here the administrator has a larger task, that is, to use the Publishing Framework plug-in to SAS Management Console to create both the channel and subscribers, subscribing individuals to the channel, and then retaining sole WriteMetadata access control for the channels and subscribers. No user can begin or end a subscription except through a request to the Admin. A publishing account is also required that would have WriteMetadata for the channel and ReadMetadata for the subscriber profiles.

Create the channel with the following permissions:

- PUBLIC – Deny ReadMetadata, Deny WriteMetadata
- Subscribed Groups – Grant ReadMetadata
- Users publishing to the channel – Grant ReadMetadata
- Administrator – Grant ReadMetadata, Grant WriteMetadata

Subscriber profiles are created by the Admin and locked with the following permissions:

- PUBLIC – Deny ReadMetadata, Deny WriteMetadata
- Users publishing to the channel – Grant ReadMetadata
- Administrator – Grant ReadMetadata, Grant WriteMetadata

Because portal channels require an archive, channel publishers must be granted WriteMetadata for channels with archives, because metadata is written to track persisted result packages in the archive.

Portal Permission Trees

Portal permission trees (folders) are maintained to secure content items that are unique to the Portal application: Pages, Portlets, Web Applications, Links, and Publication and Syndication channels. Permission trees are managed in the BI root folder, /Portal Application Tree, and contain references to Portal pages.

After an install, users can browse to this BI root folder with the Portal Navigator portlets and see their own permission tree and the permission trees of groups that they belong to.

In addition, client user interfaces that can provide the File > Save command can store content in this space. This presents two areas of possible concern:

- Some application service provider (ASP) customers might differentiate service offerings by group names to distinguish customer relationships, for example, Economy, Gold, and Platinum. They might not want customers to know this classification. Because such group names are viewable by clients, such group names should not reveal sensitive information.
- Portal users can navigate to their permission trees (user and groups that they are members of) and try to view the Pages. This is a harmless event as the Pages are already part of the user's desktop.

Portal users and groups each have a permission tree folder that is used to secure portal content. When a user or group becomes active in the Portal, that permission tree folder is created as a sub-folder in the 'Portal Application Tree' root folder.

An access control of grant WriteMetadata is needed for Portal users for the 'Portal Application Tree' root folder when a user first logs on to the Portal. When the user's permission tree is created, Portal code applies direct access controls to the tree folder as follows:

- PUBLIC – Deny ReadMetadata, Deny WriteMetadata
- Owner, either user or group – Grant ReadMetadata, Grant WriteMetadata
- Portal Admins – Grant ReadMetadata, Grant WriteMetadata, grant-Delete

These ACEs allow Portal code to access the permission tree folder.

For strongly secured Portal applications where new users must be approved for access, you can control WriteMetadata for the 'Portal Application Tree' by using a direct ACE.

Note: A direct ACE that denies WriteMetadata to SASUSERS prevents new users from first-time log on to the Portal. To enable a new user to log on and create the required user permission tree, apply a direct ACE to that user that grants WriteMetadata, then remove that direct ACE after the user's first log on.

Audits for Metadata Permissions

Customers who have strict security requirements will want to audit SAS Metadata-based security. Requirements might include:

1. Personal content is secured to the owner.
2. Group content is secured to group members. Such groupings can be based on external customer-client relationships or internal user groups.
3. Privacy: Users must not be aware of other users of the application, individually or by group association as exposed by Metadata Navigator-type clients or Search features.
4. If the customer is an application service provider (ASP), users must not be able to see infrastructure metadata about the service provider relationship. For example, a service user has been placed in a “low-priority” usage group. This information should not be available to the service user.

Here are some alternatives for performing such audits to verify compliance for security requirements, but none of them are automated for SAS 9.1.3:

- Log on to the Portal with representative role-based accounts, and use the Tree Portal Navigator portlets to navigate all available Information Service repositories and to inventory available group folders. The Tree Navigator portlet shows all content types. This makes it useful for audit checks, but the Administrator can choose to limit its availability to portal administrators.
- Log on to the Portal with representative role-base accounts, and use the Search feature, all content, and review the list. The location that is provided in the results list indicates the metadata repository location for the item.
- Log on to SAS Management Console with representative role-based accounts, and use the Authorization Manager plug-in to view sub-folders and available content.
- Review the shared content areas for products (such as SAS Web Report Studio) using representative role-based accounts. Confirm that group folders are only accessible by group members.

Areas for Customer Attention

A default installation of SAS Enterprise BI requires additional configuration by the SAS Administrator to provide adequate security for content that is managed by metadata. For specific background and guidance, see the *SAS® 9.1.3 Intelligence Platform: Security Administration Guide* available at support.sas.com/documentation/configuration/bisecag.pdf.

Note: The *SAS 9.1.3 Intelligence Platform: Security Administration Guide* includes content formerly in the *SAS 9.1.3 Enterprise Intelligence Platform: Administration Guide* before November 2006.

SAS®9 applications navigate through the metadata repository as a means to access content items. Portal Navigator portlets and Visual Data Explorer File > Open/Save dialogs will enable users to navigate the entire Information Service. This includes the content of the following BI root folders:

- /BIP Tree
- /Samples
- /Integration Technologies
- /Portal Application Tree
- additional BI root folder created by the customer

For Portal Navigator portlets, the DAV repository that is associated with the Information Service is accessible and should be secured by using the appropriate DAV administration tools.

The following review provides a quick summary of the common content folders in a SAS Enterprise Business Intelligence deployment and the first steps to provide basic metadata security.

In SAS Management Console, an Administrator might need to use a specific Manager plug-in to create a folder or content object, and then use the Authorization Manager plug-in to apply access permissions.

Administrators will find it useful to create ACTs for common access control patterns:

“Public Read Metadata Only” ACT

- PUBLIC – Grant ReadMetadata, Deny WriteMetadata
- Administrator groups (including SAS Web Administrator) – Grant ReadMetadata, Grant WriteMetadata

“Admin Access Only” ACT

- PUBLIC – Deny ReadMetadata, Deny WriteMetadata
- Administrator groups – Grant ReadMetadata, Grant WriteMetadata

“Group Owner” ACT

- PUBLIC – Grant ReadMetadata, Deny WriteMetadata
- Group – Grant ReadMetadata
- Group administrators and content authors – Grant ReadMetadata, Grant WriteMetadata
- Administrator groups – Grant ReadMetadata, Grant WriteMetadata

The Default ACT must provide ReadMetadata to either PUBLIC or SASUSERS depending on security policy. The PUBLIC group represents all authenticated users. The SASUSERS group consists of authenticated users who have User identities established.

A typical, secure access permission pattern would be applied as follows:

/BIP Tree

- Apply “Public Read Metadata Only” ACT

/BIP Tree/ReportStudio/Shared/Reports – SAS Web Report Studio Shared reports folders

- Create a “Public” sub-folder and apply ACE to grant ReadMetadata, WriteMetadata to PUBLIC
- Create group sub-folders and apply “Group Owner” ACT

/BIP Tree/ReportStudio/Maps – SAS Web Report Studio Maps folders

- Use the same sub-folder and permission pattern as used for the preceding Reports folders to provide each group secure access to group maps.

/BIP Tree/ReportStudio/Users – SAS Web Report Studio user folders

- SAS 9.1.3 Service Pack 4: Grant Admin group (includes SAS Web Administrator, SAS Web Report Studio privileged account) WriteMetadata
- SAS 9.1.3 Service Pack 3 (without hot fix 21WRS01 applied): Apply ACE to grant WriteMetadata to PUBLIC. This enables users to create and secure their own folders.

/BIP Tree/Users – Visual Data Explorer user folders.

- In SAS 9.1.3 Service Pack 4 and SAS 9.1.3 Service Pack 3 with hot fix 913CDD02 applied: User folders are secured to owner. Users cannot create content at this location. Apply ACE to grant WriteMetadata to PUBLIC, depending on whether the security policy allows creation of user folders.
- SAS 9.1.3 Service Pack 3 (without hot fix 913CDD02 applied): Folders are not secured when created.

/BIP Tree/SASGeneratedMaps

- When Visual Data Explorer or SAS Web OLAP Viewer for Java access Cubes directly, information maps are generated and stored in this location. Hot fix 913CDD01 provides an option to control direct access of Cubes.
- Generated maps can be accessed by any user who has been granted WriteMetadata permission on this location.
- Applying an ACE to deny WriteMetadata to PUBLIC or SASUSERS prevents the storage of generated maps (and direct access to cubes).

/Samples

- Apply “Admin Access Only” ACT to Grant Admin group (includes SAS Web Administrator, SAS Web Report Studio privileged account) WriteMetadata
- Grant ReadMetadata, WriteMetadata to Stored Process writers
- Apply ACEs to Grant ReadMetadata to power user groups as needed.

/Integration Technologies

- Apply “Public Read Metadata Only” ACT
- Grant ReadMetadata, WriteMetadata as needed to lower folders to manage group channels and subscriber profiles.

/Integration Technologies/Publish-Subscribe/Channels/

- Publishing Framework Channel definitions
- Admin uses the Publishing Framework plug-in to SAS Management Console to create Channels. Admin applies ACEs or ACTs to Grant groups ReadMetadata, WriteMetadata to individual Channels to see (ReadMetadata) and allow group subscription (WriteMetadata). Security and privacy needs would determine if ReadMetadata and WriteMetadata should be granted as a pair.
- (Optional) The Metadata administrator could create a channel folder and grant a channel administrator group WriteMetadata to create Channels as needed in the folder.
- If Channels have archives, apply ACEs to grant ReadMetadata, WriteMetadata to publishing groups for the channel or channel folder.

/Integration Technologies/Publish-Subscribe/Subscribers/Content Subscribers/

- Portal, SAS Enterprise Guide, and the Publishing Framework plug-in to SAS Management Console store subscriber profiles here.
- If the Admin wants sole control of subscription profiles, apply ACT to grant Admin groups ReadMetadata, WriteMetadata – Deny PUBLIC WriteMetadata (this permission pattern can be inherited from ACT applied at root folder).
- Additional access control can be provided by creating subscriber profiles and granting ReadMetadata, WriteMetadata to a limited group: owner, admin group, and publishing group.

/Portal Application Tree - Portal permission trees for content items

- Portal Users require Grant ReadMetadata, Grant WriteMetadata for this folder. This is usually inherited from the Default ACT. Permission trees that are created for users and groups are secured to the owning user or group.
- Admins, including the SAS Web Administrator account, must also have Grant ReadMetadata, Grant WriteMetadata for this root folder.
- The SAS Web Administrator account is used to manage Portal permissions trees at this location, and administrators might need to directly manage permission tree folders here.

Best Practices for a Stronger Metadata-based Security Policy

Limit content locations to a small number of BI root folders, preferably only in the “BIP Tree” root folder.

Limit use of Tree Navigator portlet to administrative users only: To do so, complete the following steps:

1. Start the SAS Management Console.
2. In the Authorization Manager, select Resource Management, By Type, Prototype folder.
3. Locate “TreeNavigator template”.
4. Select Properties.
5. On the Authorization tab, Grant ReadMetadata to Admin groups, Deny ReadMetadata to PUBLIC.

Deny WriteMetadata access whenever possible. This prevents users from writing content in unsecured locations.

Generally, ReadMetadata must be broadly granted for navigation and searching, and can be denied at the lowest subfolders that must be secured to the owning user or group.

When applying a direct ACE or ACT to deny ReadMetadata permission for an identity, always deny WriteMetadata as well for the identity.

Appendix A – Flat Folder Structure Permissions – Generic Example

The specific location of content items will depend on the application and the customer's environment. The following is a generic example that shows access control permission patterns for Information Maps in a typical Web Report Studio installation.

The Information Architect has identified a set of information maps that must be available only to a certain group, and another set to be available only to another group. As an example, one group, Dept A, may be decision makers viewing Maps with the Data Explorer in the Information Delivery Portal, while the other group, Dept B, could be business analysts working with Web Report Studio to deliver web reports. Due to the sensitive information exposed by the Maps, access must be limited to specific groups.

Default ACT:

- PUBLIC – Deny ReadMetadata, Deny WriteMetadata
- SASUSERS – Grant ReadMetadata, Grant WriteMetadata

Note: SAS application users must have a User identity in the metadata repository to access content.

Web Report Studio (SAS 9.1.3 Service Pack 3) requires that Maps be stored in a specific location in the Metadata repository:

/BIP Tree/ReportStudio/Maps/

Administrator creates an Administrators group that contains SAS Web Administrator account and other customer administrator accounts. Our generic user groups are called Dept A and Dept B. The Administrator also creates a group called "DW Analysts" consisting of programmers and analysts working with the data warehouse and information maps.

Common content location:

/BIP Tree/ReportStudio/Maps/

Apply default permissions using ACE:

- PUBLIC –Deny Write Metadata
- Administrator groups – Grant WriteMetadata
- DW Analysts – Grant WriteMetadata

Result: Blocks all users except Admin groups and Analysts from creating folders or adding content at this location.

Admin creates

/BIP Tree/ReportStudio/Maps/DeptA/

Apply "Dept A Owner" permissions using ACE:

- PUBLIC – Deny ReadMetadata, Deny WriteMetadata
- Dept A – Grant ReadMetadata
- Administrator groups – Grant ReadMetadata, Grant WriteMetadata
- DW Analysts – Grant ReadMetadata, Grant WriteMetadata

Result: Only Dept A group members, Admin groups and Analysts can access this folder to view reports. Only Admins and Analysts can create or manage maps.

/BIP Tree/ReportStudio/Maps/DeptB/

Apply "Dept B Owner" permissions using ACE:

- PUBLIC – Deny ReadMetadata, Deny WriteMetadata
- Dept B – Grant ReadMetadata
- Administrator groups – Grant ReadMetadata, Grant WriteMetadata
- DW Analysts – Grant ReadMetadata, Grant WriteMetadata

Result: Only Dept B group members, Admins, and Analysts can access this folder to view maps. Only Admins and Analysts can create or manage maps.

Note: deny PUBLIC WriteMetadata and SAS Admins Grant WriteMetadata are inherited from the Reports parent folder.

You can define the above collection of ACEs as an Access Control Template (ACT) that you apply to the group folder to secure a group folder.

Complete the following steps to create the ACT:

1. Start the SAS Management Console.
2. In the Authorization Manager plug-in, select Properties for the group folder, then the Authorization tab.
3. Click the Access Control Template button to apply the ACT to the folder. ACEs applied through a direct ACT will appear with a green background for the permission check boxes viewed in the Properties, Authorization tab dialog.

Appendix B - Hierarchic Folder Structure – Generic Example

As an example, we will look at the shared Web Reports location in the BI Tree root folder:

/BIP Tree/ReportStudio/Shared/Reports

By default, this location in the metadata repository has only those access controls that are inherited from the Default ACT, meaning that most groups will have WriteMetadata at this location and can create folders and reports.

For a site where group content must be secure to the group, the first step for the Administrator will be to create group folders that represent broad user groupings at the site: Sales, Marketing, Operations, Finance, Human Resources, etc. In a secure setting, the Administrator may also limit the creation of folders at this level, to force users to work in secure subfolders created for them.

If there is a need for sharing non-secured reports between groups, the administrator can create an open access Public folder and allow Web Report Studio users to create folders and save reports in these folders.

Example:

/BI Tree/ReportStudio/Shared/Reports/Public
/BI Tree/ReportStudio/Shared/Reports/Sales
/BI Tree/ReportStudio/Shared/Reports/Marketing
/BI Tree/ReportStudio/Shared/Reports/Finance
/BI Tree/ReportStudio/Shared/Reports/Executive

Use Case Scenario

A U.S. wholesale business divides Sales territories into regions: Southeast, Southwest, Northeast, Northwest, and then by States within regions. Sales teams are managed by State, with a State manager and a Regional manager. Four regional managers report to the Sales Executive. Sales reports include discounts and commission data so State managers must not see other managers' reports. Regional managers can review all State reports for their region but not other region's reports. The Sales Exec can review regional and state reports, and shares a US Sales report with the company's Executive group.

Administrator creates and populates Metadata groups as follows:

Admins – SAS Administrator(s)
BI Analysts – BI Content creator(s)
Executive – Sales Exec is a member, with others
Regional Sales Managers – 4
State Managers by Region – 4 groups
 Southeast State Mgrs
 Northeast State Mgrs
 Southwest State Mgrs
 Northwest State Mgrs
State Sales Managers = 4 Regional State Manager groups

The hierarchic folder structure looks like this:

```
../Reports/Public
../Reports/Executive
../Reports/Sales
../Reports/Sales/National
../Reports/Sales/Southeast/Region
../Reports/Sales/Southeast/Florida
../Reports/Sales/Southeast/Georgia
...
```

When building permissions in a hierarchic folder structure, there are two approaches:

1. Provide broad access at the top of the hierarchy and block it as you work down the subfolders
2. Provide limited access at the top of the hierarchy and add access to subfolders as you work down the hierarchy.

In deep hierarchies, either approach can be difficult to visualize as effective permissions are combinations of inherited permissions and direct permissions. An ACT can be created for repeating patterns of permissions. Applying the ACT to each subfolder in the hierarchy can make it easier to determine effective permissions.

In this use case, the second approach of providing limited access at the top of the hierarchy and add access to subfolders will be used, combined with an ACT.

Admin begins work here:
/BIP Tree/ReportStudio/Shared/Reports/

Set permissions using ACE:

- PUBLIC –Deny WriteMetadata
- Administrator groups – GrantWriteMetadata

Result: Only Admins can create folders or content at this folder. This prevents WRS users from accidentally saving reports in an unsecured location.

Admin creates
/BIP Tree/ReportStudio/Shared/Reports/Public/
with a Direct ACE to grant WriteMetadata for SASUSERS. This creates a public location (for registered users) for sharing non-sensitive content.

Admin uses SAS Management Console Authorization Manager to create an ACT called “Base Sales” as follows:

- PUBLIC – Deny ReadMetadata
- Administrator groups – Grant ReadMetadata, GrantWriteMetadata
- BI Analysts – Grant ReadMetadata, GrantWriteMetadata
- Executives – Grant ReadMetadata

Result: Applying this ACT to a folder (or any object) blocks inheritance of ReadMetadata for all users, and allows Admins to administer content, BI Analysts to navigate and create content, and Executives to navigate and view content.

Admin creates
../Shared/Reports/Sales

Permissions:

Apply Base Sales ACT to set the following permissions:

- Regional Sales Managers group – Grant ReadMetadata
- State Sales Managers group (a supergroup of groups) – Grant ReadMetadata

Result: Admins, BI Analysts, and all Sales management can navigate to this folder. Other employees cannot navigate to this folder. BI Analysts and Admins can manage folders at this level.

Admin creates
../Shared/Reports/Sales/National

Permissions:

Apply Base Sales ACT

Result: Executives only can view this folder and content. Admins and Analysts can create and manage content.

Admin creates
../Shared/Reports/Sales/Southeast

Permissions:

Apply Base Sales ACT

Create a direct ACE to set the following permissions:

- Southeast Region Sales Manager – Grant ReadMetadata
- Southeast State Sales Managers – Grant ReadMetadata

Result: Execs and Southeast regional manager and Southeast state managers can navigate to this folder.

Admin creates
../Shared/Reports/Sales/Southeast/Region

Permissions:

Apply Base Sales ACT

Create a direct ACE to set the following permissions:

- Southeast Region Manager – Grant ReadMetadata

Result: Execs and Southeast Regional manager can read content in this folder. Admins and Analysts can create and manage content.

Admin creates
../Reports/Sales/Southeast/Georgia

Permissions:

Apply Base Sales ACT

Create a direct ACE to set the following permissions:

- Georgia Sales Manager – Grant ReadMetadata
- Southeast Region Manager – Grant ReadMetadata

Result: Execs and Southeast Regional Sales Manager and Georgia Sales Manager can view this folder and content. Admins and Analysts can create and manage content.

Admin creates
../Reports/Sales/Southeast/Florida

Permissions:

Apply Base Sales ACT

Create a direct ACE to set the following permissions:

- Florida Sales Manager – Grant ReadMetadata
- Southeast Region Manager – Grant ReadMetadata

Result: Execs and Southeast Regional Sales Manager and Florida Sales Manager can view this folder and content. Admins and Analysts can create and manage content.

Folder creation pattern repeats through each region.