SAS® 9.2
Management Console
Guide to Users and Permissions

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2009. *SAS®) 9.2 Management Console: Guide to Users and Permissions.* Cary, NC: SAS Institute Inc.

SAS® 9.2 Management Console: Guide to Users and Permissions

ISBN 978-1-59994-585-9

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st electronic book, February 2009
1st printing, March 2009

Contents

Chapter 1
Concepts

About This Document

This document helps you administer users and permissions in SAS Management Console. It explains key concepts and provides step-by-step instructions for selected tasks. For more information about security, see the *SAS Intelligence Platform: Security Administration Guide*.

Introduction to User Administration

About User Administration

In order to make access distinctions and track user activity, security systems must know who is making each request. The primary purpose of user administration is to provide information that helps systems make this determination. The central piece of user information that the SAS environment requires is one external account ID for each user. The SAS environment uses its copy of these IDs to establish a unique SAS identity for each connecting user. All of a user's group memberships, role memberships, and permission assignments are ultimately tied to their SAS identity.

Note: For identification purposes, only the account IDs are needed. SAS doesn't maintain copies of external passwords for identification purposes.

To access user administration features in SAS Management Console, select the **User Manager** node on the **Plug-ins** tab. Your roles and permissions determine which user administration tasks you can perform.

Note: Don't confuse the **Users** folder (on the **Folders** tab) with the **User Manager** node (on the **Plug-ins** tab). The **User Manager** node is the only location from which you can manage identities. The **Users** folder provides containers in which users can store their content.

TIP As an alternative to interactively creating and maintaining identity information, you can write a program that performs these tasks as batch processes. See the user import macros documentation in the *SAS Intelligence Platform: Security Administration Guide*.

About Users

A user is an individual person or service identity.

We recommend that you create an individual SAS identity for each person who uses the SAS environment. This enables you to make access distinctions in the metadata layer and establishes a personal folder for each user. If generic access is sufficient for some of your users, those users can instead share the generic PUBLIC group identity.

An individual SAS identity is established by coordination between two sets of identity information:

- in an external system, a user account
- in the metadata, a user definition that includes a copy of the external account ID

To give someone an individual SAS identity, you create a metadata user definition that includes a copy of their external account ID. This list provides details for several configurations:

- In the simplest configuration, each user needs an account that is known to the metadata server's host.
  - If the metadata server is on Windows, users typically have Active Directory accounts.
  - If the metadata server is on UNIX, users might have UNIX accounts. Sometimes a UNIX host recognizes LDAP, Active Directory, or other types of accounts.
- In a common alternate configuration, the metadata server trusts authentication that is performed at the Web perimeter. In this configuration, anyone who uses a Web application needs a Web realm account.
- In a less common alternate configuration, the metadata server directly uses an LDAP provider such as Active Directory. This is appropriate only if you have accounts that aren't already accepted by the metadata server's host. For example, if the metadata server is on Windows, it isn't necessary (or appropriate) to configure direct use of Active Directory.

Note: A PUBLIC-only user doesn't need a metadata user definition (but does need an account). For metadata administrators and some service identities, it is appropriate to use a SAS internal account.

About Groups

A group is a set of users.

We recommend that you create groups to simplify security management as follows:

- It is more efficient to assign permissions to groups than to individual users.
- If you need to store passwords in the metadata, you can reduce the amount of required maintenance by using a group to make one shared account available to multiple users.
- It is sometimes more efficient to manage role membership by assigning groups to roles instead of assigning users directly to roles.

This table introduces three predefined groups:

Table 1.1 *PUBLIC, SASUSERS, and SAS Administrators*

Group	Description
PUBLIC	Includes everyone who can access the metadata server (directly or through a trust relationship).
SASUSERS	Includes those members of the PUBLIC group who have a well-formed user definition.
SAS Administrators	Should include only users who perform metadata administrative tasks. In a standard configuration, members are granted broad access but aren't unrestricted.

TIP A group's membership can include other groups as well as individual users. This enables you to create a nested group structure.

About Roles

A role manages the availability of application features such as menu items.

An application feature that is under role-based management is called a capability. Anyone who is a member of a role has all of that role's capabilities. This list highlights key points:

- Roles determine which user interface elements (such as menu items or plug-ins) you see when you use an application. Roles don't protect data or metadata (other than a few system items).
- Having a certain capability is not an alternative to meeting permission requirements. Permission requirements and capability requirements are cumulative.
- Roles and groups serve distinct purposes. You can't assign permissions to a role or capabilities to a group.
- Capabilities are always additive. Assigning someone to a role never reduces what that person can do.

Each application that supports roles offers a fixed set of capabilities. You can't convert an application feature that is not a capability into a capability. However, if you add custom plug-ins (in SAS Management Console) or custom tasks (in SAS Enterprise Guide or the SAS Add-In for Microsoft Office) you can register those features as capabilities.

Each application that supports roles provides one or more predefined roles. Each predefined role has a unique initial set of capabilities. The capabilities that a role provides should reflect the activities and responsibilities of that role's members. You can adjust the distribution of capabilities in these ways:

- Change role memberships. For example, to prevent regular users from seeing plug-ins in SAS Management Console, you might narrow the membership of the **Management Console: Content Management** role by making changes on that role's **Members** tab.
- Customize the initial roles-to-capabilities mapping by using any of these techniques:
  - Incrementally select or clear explicit capabilities for a role. You can't deselect capabilities for the unrestricted role.
  - Aggregate existing roles so that one or more roles contributes all of their capabilities to another role.
  - Create new roles that provide unique combinations of capabilities.

This table introduces the main administrative roles:

Table 1.2 *Main Administrative Roles*

Role	Capabilities
Metadata Server: Unrestricted	Members have all capabilities and can't be denied any permissions in the metadata environment.[*]
Metadata Server: User Administration	Members can create, update, and delete users, groups, roles (other than the unrestricted role), internal accounts, logins, and authentication domains.[**]
Metadata Server: Operation	Members can administer the metadata server (monitor, stop, pause, resume, quiesce) and its repositories (add, initialize, register, unregister, delete).[***]
Management Console: Advanced	Members can see all plug-ins in SAS Management Console (in the initial configuration).

[*] Unrestricted users are subject to denials in other authorization layers, can use only those logins that are assigned to them (or to groups to which they belong), and don't have implicit capabilities that are provided by components other than the metadata server.

[**] Restricted user administrators can't update identities for which they have an explicit ☑ or ACT ✅ (green) denial of WriteMetadata.

[***] Only someone who has an external user ID that is listed in the adminUsers.txt file with a preceding asterisk can delete, unregister, add, or initialize a foundation repository. Only an unrestricted user can analyze and repair metadata or perform tasks when the metadata server is paused for administration.

About Logins

What is a Login?

A login is a SAS copy of information about an external account. Every login must include a user ID. In a login for a Windows account, the ID must be qualified (for example, *userID@company.com)*, *domain\userID*, or *machine\userID*.

TIP The requirement to provide a qualified ID for a Windows account applies to the SAS copy of the ID. It is usually not necessary to qualify the user ID that you provide when you launch a SAS application.

TIP If you do provide a qualified ID when you log on, you must use the same format that was used in your login. For example, Windows might accept both *WIN\me* and *Me.MyLastName@mycompany.com*, but SAS can understand only one of these qualified form (the form in which the SAS copy of the ID is stored).

Logins for Users

Each user should have a login that establishes their SAS identity. It is not necessary to include a password in this login. However, the password column always displays eight asterisks (regardless of whether a password is actually stored). For example, this is how Joe's login might look when a user administrator views Joe's **Accounts** tab:

```
DefaultAuth | WIN\Joe   | ********
```

A user might have additional logins that provide access to other systems. For example, if Joe has his own Oracle account, he might have these two logins:

```
DefaultAuth | joe       | ********

OracleAuth  | ORAjoe    | ********
```

Note: The Oracle login should include a copy of Joe's Oracle password.

If a site uses Web authentication, the requirements are different. For example, if Joe uses both Web and desktop applications at such a site, Joe might have these three logins:

```
DefaultAuth | WIN\Joe | ********

OracleAuth  | ORAjoe  | ********

web         | WEBjoe  | ********
```

Note: Like his DefaultAuth login, Joe's Web login is used only to launch clients, so there is no need to create a SAS copy of Joe's Web realm password.

Logins for Groups

Groups don't have to have logins. The main reason to give a login to a group is to make a shared account available to multiple users. A group login contains a SAS copy of the user ID and password for a shared account. For example, to provide shared access to DB2, a group might have a login that looks like this:

```
DB2Auth | sharedDB2id | ********
```

All members of the group can use this login. Since this login is for a third-party database, a copy of the DBMS account password should be stored in this login.

About Internal Accounts

What is an Internal Account?

An internal account is a SAS account that the metadata server authenticates independently, without relying on an external authentication provider such as the operating system. Use internal accounts only for administrators and some service identities. For these purposes, an internal account is an acceptable substitute for an external account with a corresponding login. For example, the SAS Administrator and the SAS Trusted User can be based on internal accounts.

Benefits of Internal Accounts

Internal accounts have these advantages:

- Internal accounts provide an alternative to creating external accounts for SAS internal purposes such as inter-process communication.
- Internal accounts can be maintenance free. You don't have to synchronize internal accounts with some other user registry. Internal accounts don't have to conform to the security policies of the rest of your computing environment. For example, even if your host security policy forces password changes every 30 days, you can retain the initial policy for internal account passwords (which is that these passwords never expire).
- Internal accounts are usable only in the SAS realm, so they reduce exposure to the rest of your security environment.

TIP You can also use an internal account to temporarily assume another user's identity for validation or troubleshooting purposes.

Limitations of Internal Accounts

Although the **Create Internal Account** button is available on all user definitions, internal accounts are not intended for regular users. Someone who has only an internal account can't do these things:

- launch a standard workspace server without interactively providing some external credentials
- participate in Integrated Windows authentication or Web authentication
- add, delete, initialize, or unregister a foundation repository

Policies for Internal Accounts

By initial policy, these server-level settings are in effect:

- Accounts don't expire and aren't suspended due to inactivity.
- Passwords must be at least six characters, don't have to include mixed case or numbers, and don't expire.
- The five most recent passwords for an account can't be reused for that account.
- There is no mandatory time delay between password changes.
- After three failed attempts to log on, an account is locked. If an account is locked because of log on failures, further log on attempts cannot be made for one hour.
- For an account that has a password expiration period, there is a forced password change on first use and after the password is reset by someone other than the account owner. By initial policy, passwords don't expire so there are no forced password changes.

Note: These settings are defined in the metadata server's omaconfig.xml file. In User Manager, you can customize some of these settings on a per-account basis.

CAUTION:

Passwords for a few required accounts (such as the SAS Administrator and the SAS Trusted User) are included in configuration files. If you change these passwords, you must also update the configuration. See the *SAS Intelligence Platform: Security Administration Guide*.

About Authentication Domains

What is an Authentication Domain?

An authentication domain is a name that facilitates the matching of logins with the servers for which they are valid. This matching is not important when you launch a client, but it is important when you access certain secondary servers such as a third-party DBMS or, in some configurations, a standard workspace server.

When Do I Need to Add an Authentication Domain?

In the simplest case, all logins and SAS servers are associated with one authentication domain (DefaultAuth). This list describes the most common reasons for using more authentication domains:

- If you use Web authentication, you might need a second authentication domain for the logins that contain Web realm user IDs.
- If you have a third-party server (such as a DBMS server) that has its own user registry, you need a separate authentication domain for that server and its logins.
- If both of the following criteria are met, you need a separate authentication domain for the standard workspace server and its logins:
  - The standard workspace server doesn't share an authentication provider with the metadata server (and can't be configured to do so).
  - You want to provide seamless individualized access to the standard workspace server.

About Passwords

Passwords in Logins

It is usually not necessary to create a SAS copy of an external password. The main reason to include a password in a login is to provide seamless access to a server that requires credentials that are different from the credentials that users initially submit. These are the most common examples:

- A third-party DBMS server usually requires a different set of credentials.
- In a multi-platform environment, the standard workspace server might require a different set of credentials.

If credentials aren't otherwise available, most applications prompt users for an appropriate user ID and password.

Passwords in Internal Accounts

Internal accounts exist only in the metadata. Each internal account includes a password. By initial policy, internal passwords don't expire.

Passwords in Configuration Files

Passwords for a few required accounts (such as the SAS Administrator and the SAS Trusted User) are included in configuration files. If you need to change these passwords, see the *SAS Intelligence Platform: Security Administration Guide*.

About External Identities

What is an External Identity?

While logins and internal accounts are involved in the log on process, external identities are not. An external identity is an optional synchronization key for a user, group, or role. If you use batch processes to coordinate SAS identity information with your primary user registry, you need external identities (such as employee IDs) to facilitate matching. This list explains the circumstances in which a user, group, or role needs an external identity:

- For a user, group, or role that you maintain interactively in SAS Management Console, no external identity is needed.
- For a user, group, or role that you maintain using batch processes, one external identity is needed.

Where do External Identities Come From?

External identities can be added in these ways:

- For a user, group, or role that is created by an import process, an external identity is added as part of that process.
- For any user, group, or role, you can interactively add an external identity on the **General** tab of their definition.

Requirement: Unique Names and IDs

Within a metadata server, these uniqueness requirements apply:

- You can't create a user definition that has the same name as an existing user definition.
- You can't create a group or role definition that has the same name as an existing group or role definition.
- You can't assign the same user ID to different users or groups. All of the logins that include a particular user ID must be owned by the same identity. This enables the metadata server to resolve each user ID to a single identity.
  - This requirement is case-insensitive. For example, you can't assign a login with a user ID of *smith* to one user and a login with a user ID of *SMITH* to another user.
  - This requirement applies to the qualified form of the user ID. For example, you can assign a login with a user ID of *winDEV\brown* to one user and a login with a user ID of *winPROD\brown* to another user.
  - This requirement can't be mitigated by associating the logins with different SAS authentication domains. For example, if one user has a login with a user ID of *smith* in DefaultAuth, you can't give any other user a login with the user ID *smith*, even if you put that login in another authentication domain.
- If you give a user two logins that contain the same user ID, the logins must be in different authentication domains. Within an authentication domain, each user ID must be unique. For example, if you give Tara O'Toole two logins that both have a user ID of *tara*, then you can't associate both of those logins with the OraAuth authentication domain. As with the previous requirement, this requirement is case-insensitive and is applied to the fully qualified form of the user ID.

Introduction to Access Management

About Access Management

Access management determines which items a user can interact with. The permissions that you set in SAS Management Console are part of a metadata-based access control system that SAS provides. These permission settings supplement protections in other layers (such as the operating system and the WebDAV). Across layers, protections are cumulative. You can't perform a task unless you have sufficient access in all layers.

CAUTION:

> **Do not rely exclusively on metadata layer permissions to protect data.** Manage physical access (operating system and DBMS permissions) in addition to metadata layer access.

You manage access to an item as part of the item's properties (on the item's **Authorization** tab). Your roles and permissions determine which access management tasks you can perform.

Granularity and Mechanics of Permissions

Repository-Level Controls

Repository-level controls function as a gateway. Participating users usually need ReadMetadata and WriteMetadata permissions for the foundation repository. Repository-level controls also serve as a parent-of-last-resort, defining access to resources that don't have more specific settings. Repository-level controls are defined on the **Permission Pattern** tab of the repository ACT.

Resource-Level Controls

Resource-level controls manage access to a specific item such as a report, an information map, a stored process, a table, a cube, or a folder. You can define resource-level controls individually (as explicit settings) or in patterns (by using access control templates).

Fine-Grained Controls

Fine-grained controls affect access to subsets of data within a resource. To establish fine-grained controls, you define permission conditions that constrain access to rows within a table or members within an OLAP dimension.

Feature-Level Controls

Some applications use roles to limit access to functionality. These applications check each user's roles in order to determine which menu items and features to display for that user. Roles are not an authorization feature; they are managed and documented as part of user administration.

Inheritance and Precedence of Permissions

Two Relationship Networks

Permission settings are conveyed across two distinct relationship networks, a resource network and an identity network. Permissions that are set directly on an item have priority over permissions that are set on the item's parent. For example, when access to a report is evaluated, a denial that is set on the report (and assigned to the PUBLIC group) overrides a grant that is set on the report's parent folder (even if the grant is assigned to you).

The Resource Relationships Network

Permissions that you set on one item can affect many other items. For example, a report inherits permissions from the folder in which the report is located. This relationship network consists primarily of a folder tree. This list highlights exceptions:

- The root folder isn't the ultimate parent. This folder inherits from the repository (through the permission pattern of the repository ACT).
- The root folder isn't a universal parent. Some system resources (such as application servers, identities, and ACTs) aren't in the folder tree so they have the repository as their immediate and only parent.
- Inheritance within a table or cube follows the data structure. For example, table columns and cube hierarchies don't have a folder as their immediate parent. Instead, a column inherits from its parent table and a hierarchy inherits from its parent cube.
- In unusual circumstances, it is possible for an item to have more than one immediate parent. If there is a tie in this network (for example, if there are no settings on an item, the item has two immediate parents, and one parent provides a grant while the other parent provides a denial), the outcome is a grant. In other words, a grant from any inheritance path is sufficient to provide access.

The Identity Relationships Network

Permissions that you assign to one group can affect many other identities. For example, if you grant a group access to an OLAP cube, that grant applies to all users who are members of the group. This relationship network is governed by a precedence order that starts with a primary identity, can incorporate multiple levels of group memberships, and ends with implicit memberships in SASUSERS and then PUBLIC. If there is a tie in this network (for example, if you directly assign a user to two groups and give one group a grant and another group a deny), the outcome is a deny.

Use and Enforcement of Each Permission

Table 1.3 Use and Enforcement of Each Permission

Permission (Abbreviation)	Actions Affected and Limitations on Enforcement
ReadMetadata (RM)	View an item or navigate past a folder. For example, to see an information map you need RM for that information map. To see or traverse a folder you need RM for that folder.
WriteMetadata (WM)	Edit, delete, change permissions for, or rename an item. For example, to edit a report you need WM for the report. To delete a report you need WM for the report (and WMM for the report's parent folder). WM affects the ability to create associations. For example, you need WM on an application server in order to associate a library to that server. WM affects the ability to create items in certain containers. For example, to add an item anywhere in a repository you need WM at the repository level. For folders, adding and deleting child items is controlled by WMM, not WM.
WriteMemberMetadata (WMM)	Add an item to a folder or delete an item from a folder. For example, to save a report to a folder you need WMM for the folder. To remove a report from a folder, you need WMM for the folder (and WM for the report). To enable someone to interact with a folder's contents but with not the folder itself, grant WMM and deny WM.*

Permission (Abbreviation)	**Actions Affected and Limitations on Enforcement**
CheckInMetadata (CM)	Check in and check out items in a change-managed area. Applicable only in SAS Data Integration Studio.[**]
Administer (A)	Operate (monitor, stop, pause, resume, refresh, or quiesce) servers and spawners. For the metadata server, the availability of similar tasks is managed by the Metadata Server: Operation role (not by this permission).
Read (R)	Read data. For example, while you need RM for a cube in order to see a cube, you need R for that cube in order to run a query against it. Enforced for OLAP data, information maps, data that is accessed through the metadata LIBNAME engine, and dashboard objects.
Create (C)	Add data. For example, on a table, C controls adding rows to the table. Enforced for data that is accessed through the metadata LIBNAME engine.
Write (W)	Update data. For example, on a table, W controls updating the rows in the table. Enforced for data that is accessed through the metadata LIBNAME engine, for publishing channels, and for dashboard objects.
Delete (D)	Delete data. For example, D on a library controls the deletion of tables from the library. Enforced for data that is accessed through the metadata LIBNAME engine and for dashboard objects.

[*] A folder's WMM settings mirror its WM settings unless the folder has explicit ☑ or ACT ✅ (green) settings of WMM. A grant (or deny) of WMM on a folder becomes an inherited grant (or deny) of WM on the items and subfolders within that folder. WMM is not inherited from one folder to another.

[**] In any change-managed areas of a foundation repository, change-managed users should have CM (instead of WM and WMM).

Note: The table server supports additional permissions. See the documentation for that component.

Chapter 2
User Administration Tasks

Getting Information About a User

What Groups is This User In?

This list explains how group memberships are displayed for a user named Joe:

direct groups
: If Joe is directly assigned to any groups, those assignments are displayed in the **Member of** list box on the **Groups and Roles** tab in Joe's Properties dialog box.

indirect groups
: If Joe is a member of a group that is a member of another group, Joe is an indirect member of the second group. Because indirect membership is not displayed in Joe's **Member of** list box, you must check the properties of each group that Joe belongs to in order to determine whether that group is a member of another group.

implicit groups
: If Joe has a well-formed user definition, he automatically belongs to both the PUBLIC and SASUSERS groups. These implicit memberships are not reflected in Joe's **Member of** list box.

What Roles is This User In?

This list explains how role memberships are displayed for a user named Joe:

direct roles
: If Joe is directly assigned to any roles, those assignments are displayed in the **Member of** list box on the **Groups and Roles** tab in Joe's Properties dialog box.

indirect roles
: If Joe is a member of a group that is assigned to a role, Joe is an indirect member of that role. Because indirect membership is not displayed in Joe's **Member of** list box, you must check the properties of each group that Joe belongs to in order to determine whether that group is a member of any roles. Remember that Joe's **Member of** list box doesn't reflect his implicit membership in SASUSERS and PUBLIC. Users get most of their non-administrative capabilities through implicit membership in these groups.

contributing roles
: If Joe is in a role that has contributing roles, Joe has the capabilities of the contributing roles. To determine whether a role has contributing roles, access the role's Properties dialog box and select the **Contributing Roles** tab.

What Can This User Do?

Which Items Can This User Access?

Joe's access is not displayed as part of his user definition. Instead, Joe's permissions for a particular item are displayed on that item's **Authorization** tab.

TIP SAS programmers can create reports that document access to resources. See the discussion of security report macros in the *SAS Intelligence Platform: Security Administration Guide*.

Which Application Features are Visible to This User?

Joe has all of the capabilities that are provided by any of his roles. This list highlights key points about a role's **Capabilities** tab:

- Some roles provide implicit capabilities, which are not displayed. For example, the ability to create users is provided by the Metadata Server: User Administration role, but there is no **Create Users** check box on the **Capabilities** tab.
- A capability that has a gray check box ☑ comes from a contributing role.
- These icons indicate the status of the items beneath a node in the tree:
  - A full tree icon indicates that all of the capabilities are assigned.
  - An empty tree icon indicates that none of the capabilities are assigned.
  - A partial tree icon indicates that some of the capabilities are assigned.

What Logins Are Available to This User?

This list explains how the logins that are available to a user named Joe are displayed:

personal logins
: Joe's personal logins are displayed on the **Accounts** tab in his Properties dialog box. Only Joe and users who have user administration capabilities can see Joe's logins.

group logins
: A login that is assigned to a group can be used by any member of that group. Because Joe's group logins are not displayed on his **Accounts** tab, you must check the properties of each group that Joe belongs to in order to determine whether any of those groups have logins.

Note: On an **Accounts** tab, logins are visible only if you have user administration capabilities or you are looking at your own user definition.

Does This User Have an External Identity?

To determine whether a user has an external identity, click the **External Identities** button on the user's **General** tab.

Does This User Have an Internal Account?

To determine whether a user has an internal account, examine the bottom of the user's **Accounts** tab. If a user has an internal account, their internal ID is listed in that location. Regular users usually don't have internal accounts.

Note: Internal accounts are visible only if you have user administration capabilities or you are looking at your own user definition.

Who Can Manage Users, Groups, and Roles?

Table 2.1 *Requirements for Managing Identities*

Task	Requirements
Create users, groups, and roles. Update or delete users, groups, and roles (other than the unrestricted role). Reset other user's passwords (in metadata).	User administration capabilities, the User Manager capability, and these permissions: • WM for the identities (to update or delete them) • WM for the software components that provide role capabilities (to change capability assignments) • WM for the repository (to add identities, logins, and related items). In the initial configuration, the SAS Administrators group meets all of these requirements.
Manage the unrestricted role	Unrestricted status. In the initial configuration, only one user (the SAS Administrator) meets this requirement.

Note: Each user can manage their own personal logins in SAS Management Console or SAS Personal Login Manager.

Note: You can delegate management of an existing identity to someone who does not have user administration capabilities. See "Delegate Management of a Group or Role" on page 27.

Add Users

To create an individual SAS identity:

1. On the **Plug-ins** tab, select **User Manager**. Make sure that you are in the foundation repository.
2. For each user:
   a. Right-click and select **New** ⇨ **User**.
   b. On the **General** tab, enter a name.
   c. On the **Accounts** tab, click **New**. In the New Login dialog box, select **DefaultAuth** and enter the user's external account ID. You can use any account (LDAP, Active Directory, host, or other) that is known to the metadata server's host.

      Note: For a Windows account, qualify the ID (for example, *WIN\myID* or *myID@mycompany.com*).

Table 2.2 *Adapted Instructions for Sites That Use Web Authentication*

Type of User	Adapted Instructions[*]
Someone who uses only Web applications	Select the Web realm authentication domain (such as **web**) instead of **DefaultAuth** and enter the user's Web realm ID.
Someone who uses both Web and desktop applications	Complete the standard instructions and also add a Web realm login.

[*] If the Web user IDs and the metadata server user IDs are identical, and the Web applications don't use a standard workspace server, it isn't necessary to follow these adapted instructions.

d. Click **OK** to save the new login (it is not necessary to include a password in this login). Click **OK** again to save the new user.

3. (Optional) You can use the **Groups and Roles** tab to make a user a direct member of another group or a role.

   Note: These users automatically belong to PUBLIC (everyone who can access the metadata server) and SASUSERS (those members of PUBLIC who have a well-formed user definition).

4. Make sure that anyone who uses Windows host credentials to access a standard workspace server has the "Log on as a batch job" right. Usually, this involves adding the user's Windows account to a Windows group that is named something like **SAS Server Users**.

5. If you need to provide seamless access to a third-party server such as a DBMS, either give the user a second login or make the user a member of a group that has a shared login for the third-party server. See “Make a SAS Copy of DBMS Credentials” on page 23.

Note: You don't have to make changes on the user's **Authorization** tab. This tab has no effect on what the user can do.

Add Administrators

To create an individual SAS identity that is based on an internal account:

1. On the **Plug-ins** tab, select **User Manager**. Make sure that you are in the foundation repository.

2. For each administrator:

   a. Right-click and select **New** ⇨ **User**.

   b. On the **General** tab, enter a name.

      Note: The administrator's internal user ID will be based on this name, so it is a good idea to use a short identifier.

   c. On the **Accounts** tab, click **Create Internal Account**. In the New Internal Account dialog box, enter and confirm an initial password.

Note: By initial policy, internal passwords must be at least six characters, don't have to include mixed case or numbers, and don't expire.

TIP If you want to force a password change on first use, set a password expiration period.

d. On the **Groups and Roles** tab, move the SAS Administrators group to the **Member of** list box. This makes the new user a member of SAS Administrators.

e. Click **OK** to save the new administrator.

3. (Optional) To verify your work, examine the SAS Administrators group:

   a. In the main display, select the **SAS Administrators** group, right-click, and select **Properties**.

   b. On the **Members** tab, verify that the new administrators are in the **Current Members** list box.

   c. On the **Groups and Roles** tab, verify that the **Member of** list box includes at least these standard memberships:

      - **Metadata Server: User Administration**
      - **Metadata Server: Operation**
      - **Management Console: Advanced**

      In a standard configuration, members of the SAS Administrators group are able to perform almost all administrative tasks.

This list highlights key points:

- You don't have to use internal accounts for your administrators. You can choose to give an administrator an external account and a corresponding login as you would for a regular user.
- When you log on with an internal account, remember to include the @saspw suffix (for example, *sasadm@saspw*).
- A few administrative tasks (such as validating a workspace server, testing prompts, performing backups, and importing and exporting physical content) use a standard workspace server. Someone who has only an internal account can't perform such tasks without interactively providing external credentials.
- If you want to make someone an unrestricted administrator, move the **Metadata Server: Unrestricted** role to the **Member of** list box in step 2d.
- To conform to the rule of least privilege, we recommend that administrators do not also serve as regular users. If you want someone to be an administrator only some of the time, create two user definitions for that person.

  - One definition is based on an external account and is not a member of SAS Administrators.
  - The other definition is based on an internal account and is a member of SAS Administrators.

  A dual user logs on with their internal account when they need administrative privileges and with their external account the rest of the time.

Manage Passwords

Passwords for a few required accounts (such as the SAS Administrator and the SAS Trusted User) are included in configuration files. If these passwords change, you must also update the configuration. See the *SAS Intelligence Platform: Security Administration Guide*.

Update the Password in a Login

Password management for logins is driven by changes that occur in other systems. For example, if you have a personal login for a third-party DBMS, and you change your DBMS password, you must also update the SAS copy of that password.

Note: Most logins don't include passwords, so this is not an extremely common task. Typically, each user updates their own logins as necessary in SAS Management Console or SAS Personal Login Manager.

1. Select the user (or group) whose external password has changed.
2. Right-click and select **Properties**.
3. On the **Accounts** tab, select the login that you need to update and click **Edit**.

   Note: The password column always displays eight asterisks. Don't interpret the presence of the asterisks as an indication that a password is stored.

   Note: On the **Accounts** tab, logins are visible only if you have user administration capabilities or you are looking at your own user definition.
4. In the Login Properties dialog box, enter and confirm the new password.

Reset an Internal Password

Note: Typically, each administrator updates their own internal password as necessary (in SAS Management Console or SAS Personal Login Manager).

1. Select the user whose internal password you want to reset.
2. Right-click and select **Properties**.
3. At the bottom of the user's **Accounts** tab, click **Update**.

   Note: If this button is not present, the user doesn't have an internal account. Typically, only administrators and some service identities have internal accounts.
4. In the user's Internal Account Properties dialog box, enter and confirm a new password.

   Note: By initial policy, internal passwords must be at least six characters, don't have to include mixed case or numbers, and don't expire.
5. If you are resetting someone else's password, inform the owner of the account that their password has been reset and tell them what the new password is.

Note: By initial policy, the owner of the account is forced to change the password on first use following a password reset. This policy applies only to accounts that have a password expiration period. This policy doesn't apply when you reset your own password.

Add Contact Information

Some application features (such as subscriptions to publishing channels) can use contact information that is stored in user definitions.

1. Select the user whose phone number, e-mail address, or location you want to store.

   Note: You can't store contact information for groups or roles.

2. Right-click and select **Properties**.
3. On the **General** tab, select the **Email**, **Phone**, or **Address** tab and then click **New**.
4. In the Properties dialog box, enter contact information.

   TIP If you batch synchronize users and want to preserve contact information that you enter interactively, use a consistent value in the **Type** field. In your synchronization code, you can use this value to define exceptions that exclude this data from the batch update.

Create a Custom Group

Why Create a Custom Group?

Most predefined groups are either very broad (PUBLIC, SASUSERS) or very narrow and highly privileged (SAS Administrators). Create more groups for these reasons:

- To manage permissions for distinct classes of access. For example, you might create a group for each business unit or functional area of responsibility.
- To make a shared credential available to multiple users. See "Store Shared Credentials for a DBMS" on page 23.

How to Create a Custom Group

1. On the **Plug-ins** tab, select **User Manager** and make sure you are in the correct repository.

   Note: You usually create groups in the foundation repository. You can also create groups in custom repositories.

2. Right-click and select **New** ⇨ **Group**.
3. In the Properties dialog box:

   a. On the **General** tab, enter a name.

   b. On the **Members** tab, assign user or groups to the new group.

   c. If you want to make this group a member of other groups or roles, use the **Groups and Roles** tab.

   d. If you are using this group to make a shared account available, add a shared login on the **Accounts** tab.

Note: You don't have to make changes on the group's **Authorization** tab. This tab has no effect on what the group can do.

Create a Custom Role

Why Create a Custom Role?

In many cases, the predefined roles are sufficient. You might choose to create additional roles for these reasons:

- To decrease the level of granularity by creating an umbrella role that aggregates two or more existing roles. For example, you might create a role that includes all capabilities other than those of the most privileged roles.
- To increase the level of granularity by creating a mini-role that provides only a subset of the capabilities of a predefined role. For example, you might create a custom role called Report Distribution that provides only the report scheduling and distribution capabilities for SAS Web Report Studio.
- To create a cross-application role for a particular type of functionality. For example, you might create an OLAP role that includes the OLAP capabilities from SAS Enterprise Guide and the SAS Add-In for Microsoft Office.

How to Create a Custom Role

1. On the **Plug-ins** tab, select **User Manager** and make sure you are in the correct repository.

   Note: You usually create roles in the foundation repository. You can also create roles in custom repositories.

2. Right-click and select **New** ⇨ **Role**.
3. In the Properties dialog box:
   a. On the **General** tab, enter a name.

   b. On the **Members** tab, assign users and groups to the role.

   c. Define the role's capabilities using either or both of these techniques:

      - Assign capabilities to the role by selecting check boxes on the **Capabilities** tab. Clicking a tree icon changes the status of the selections beneath that icon's node.
      - Give this role all of the capabilities of one or more other roles by using the **Contributing Roles** tab. For example, to create a role that includes all capabilities other than those of the most privileged roles, select the **Contributing Roles** tab, move all roles over and then move the metadata server roles back.

        Note: Changes that you make to a role's capabilities affect any roles to which that role contributes its capabilities.

        Note: You can't selectively assign or incrementally remove a contributed capability.

Note: You don't have to make changes on the role's **Authorization** tab. This tab has no effect on what the role can do.

TIP (Optional) To test the new role, temporarily assume the identity of one of its members. See "Assume Another User's Identity" on page 26.

Change a Role's Capabilities

CAUTION:

There is no automated method for reverting a role back to its original set of capabilities. The initial capabilities-to-roles mapping is appropriate in many cases. Instead of adjusting the capabilities of a predefined role, consider creating a new role.

To change the set of capabilities that a role provides:

1. Make sure you have a current backup.
2. In **User Manager**, select the role.
3. Right-click and select **Properties**.
4. Use either or both of these techniques:

   - Incrementally add or remove capabilities from the role by selecting or clearing check boxes on the **Capabilities** tab.

     Note: A capability that has gray shading behind its check box ☑ comes from a contributing role and can't be removed individually.

     Note: If you click a selected white check box (because you want to clear that check box) and you then see a selected gray check box ☑, your removal of the explicit assignment has revealed an underlying contributed capability.

     Note: You can't deselect capabilities for the unrestricted role.

   - Give the role the capabilities of one or more other roles by using the **Contributing Roles** tab.

     Note: These relationships are dynamic; changes that you make to a role's capabilities affect any roles to which that role contributes its capabilities.

     Note: These relationships are monolithic; you can't selectively assign or incrementally remove a contributed capability.

   This list provides details about the **Capabilities** tab:

   - Some roles include implicit capabilities, which are not displayed on this tab. For example, the ability to create users is part of the Metadata Server: User Administration role but there is no **Create Users** check box on the **Capabilities** tab.
   - The tree icons indicate the status of the items beneath a node in the tree. Clicking a tree icon changes the status of the selections beneath that icon's node. The status cycles between full, empty, and partial states, with these exceptions:

     - The empty state doesn't occur if there are contributed capabilities.
     - The partial state occurs only if the original settings were mixed (some capabilities selected, some capabilities not selected).

       Note: The original settings are a cache of the selections that were in place at the time that you first click a particular tree icon. Any intervening action (such as clicking a check box or clicking the tree icon for a different node) causes an update to the original settings cache. There is no cache of earlier states. If you want to undo all of your changes, click **Cancel**.

5. (Optional) On the **General** tab, update the role's description to reflect its revised capabilities.
6. Click **OK** to save the changes to the role.
7. (Optional) To test the role, temporarily assume the identity of one of its members. See "Assume Another User's Identity" on page 26.

Adjust Group or Role Membership

1. In **User Manager**, select the group or role whose membership you want to change.
2. Right-click and select **Properties**.
3. On the **Members** tab, add or remove identities from the group or role.

   Note: The **Current Members** list box displays only direct members.

   Note: You can't make a role a member of a group or of another role. You can instead make one role contribute all of its capabilities to another role.

   Note: On a group definition, don't confuse the **Members** tab with the **Groups and Roles** tab. Use a group's **Groups and Roles** tab only if you want to make that group a member of other groups or roles.

   TIP You can filter the contents of the **Available Members** list box by using the **Search** radio button and the **Show Users**, **Show Groups**, and **Search All Repositories** check boxes.

Make a SAS Copy of DBMS Credentials

To provide seamless access to a third-party DBMS, add a login that contains the user ID and password for a DBMS account. These instructions are also appropriate for providing seamless access to other servers that require credentials that are different from the credentials with which a user initially logs on.

Store Shared Credentials for a DBMS

1. Verify the authentication domain for the DBMS:

   a. On the **Plug-ins** tab, expand the **Server Manager** node and select the DBMS server.

   b. Right-click, select **Properties**, and access the **Options** tab.

2. In **User Manager**, identify or create the group that you will use to manage the shared DBMS account that you want to share. For example, if you want all users to share the account, use the PUBLIC group.
3. On the group's **Accounts** tab, click **New**.
4. In the New Login Properties dialog box:

   a. Enter the user ID and password for the DBMS account.

   b. Select the authentication domain that you saw in step 1b.

   c. Click **OK** to save the login.

5. On the group's **Members** tab, make sure that everyone who needs to use the shared account is a member. Remember that only direct memberships are displayed, but indirect or implicit membership is also sufficient for making the credentials available.

Store Individual Credentials for a DBMS

Follow the instructions in the preceding topic but add the login to a user's **Accounts** tab instead of a group's **Accounts** tab.

Note: If a user has more than one available login in a particular authentication domain, the login that is closest to the user is used. If there is tie (for example, if a user is a direct member of two groups and both groups have logins in the same authentication domain), the same login is used consistently but you can't control which of the two logins is used.

Unlock an Internal Account

By initial policy, three consecutive failed attempts to log on with an internal account locks that account for one hour. To unlock a locked internal account:

1. In **User Manager**, select the user whose internal account is locked. Right-click and select **Properties**.
2. Select the **Accounts** tab. In the confirmation message box, click **Yes**.

Adjust Policies for an Internal Account

You can use per-account settings to selectively override some of the server-level policies for internal accounts.

TIP To verify the current server-level settings, examine the metadata server's omaconfig.xml file.

1. On the user's **Accounts** tab, click **Update** to open the Internal Account Properties dialog box.
2. Make changes in the **Custom Settings** group box.

   Note: There are two distinct expiration settings. Don't confuse the account expiration date with the password expiration period.

   TIP A few required accounts (such as the SAS Administrator and the SAS Trusted User) are included in configuration files. To minimize administrative effort, don't add expiration dates to these accounts or expiration periods to these passwords.

Manage Authentication Domains

Add an Authentication Domain

1. On the **Plug-ins** tab, select **User Manager** (or **Server Manager**).
2. Right-click and select **Authentication Domains**.

   Note: This menu item is available only if you have user administration capabilities.
3. In the Authentication Domain Management dialog box, click **New**.
4. In the New Authentication Domain dialog box, enter a name.

Rename an Authentication Domain

CAUTION:

Changing the name of an authentication domain can interfere with single sign-on. Do not rename an authentication domain unless you need to make a correction. In particular, avoid renaming DefaultAuth because this requires that users update their connection profiles to use the new name.

This list explains how users can access their connection profiles:

- In most Java desktop clients, select **File** ⇨ **Connection Profile** and then click **Edit**.
- In SAS Enterprise Guide, select **Tools** ⇨ **Options** ⇨ **Administration** ⇨ **Modify** and then click **Modify**.
- In the SAS Add-In for Microsoft Office, select **SAS** ⇨ **Tools** ⇨ **Connections** and then click **Modify**.

Note: If Web applications reuse initial logon credentials, you must also update and redeploy the Web applications.

To change the name of an authentication domain, select a row in the Authentication Domain Management dialog box and click **Edit**.

Delete an Authentication Domain

CAUTION:

When you delete an authentication domain, all of the logins in that authentication domain are deleted. Before you delete an authentication domain, make sure you have a current backup.

To delete an authentication domain, select a row in the Authentication Domain Management dialog box and click **Delete**.

Rename a User, Group, or Role

CAUTION:

Do not change the name of a predefined role. If you change the name of a user who has an internal account, that user's internal ID changes too. We recommend that you avoid changing identity names and instead add or update display names.

1. In **User Manager**, select the user, group, or role that you want to rename.
2. Right-click and select **Properties**.
3. On the **General** tab, add or edit text in the **Display Name** field. For an identity that doesn't have a display name, the name serves as the display name.

Delete a User, Group, or Role

CAUTION:

When you delete a user, group, or role, you lose all of that identity's associations (such as permission settings and memberships). Creating a new identity with the same name does not restore the associations.

1. Select the user, group, or role that you want to delete.
2. Right-click and select **Delete**. In the confirmation message box, click **Yes**.

Assume Another User's Identity

If you have user administration capabilities, you can temporarily add an internal account to another user's definition. This enables you to log on as that user, which can be useful for validating changes or troubleshooting problems.

1. On the users's **Accounts** tab, click **Create Internal Account** and then enter and confirm a password.

   Note: By initial policy, internal passwords must be at least six characters, don't have to include mixed case or numbers, and don't expire.

   TIP To ensure that you won't have to change the password on first use, set the account password to never expire.

2. Use the internal account ID and password to log on to a SAS application and see what access and features the user has.

   Note: Because you aren't using the user's external account ID, this technique won't reproduce a problem that is caused by the user not having a well-formed definition.

   Note: You can't use an internal account to log on to a Web application that uses Web authentication.

3. When you are finished, return to the user's **Accounts** tab and click **Delete** to remove the internal account.

Delegate Management of a Group or Role

To delegate management to someone who does not have user administration capabilities, use explicit ☑ or ACT ☑ (green) grants of the WriteMetadata permission. For example, to delegate management of a group named ETL Developers to a user named Tara, you would access the **Authorization** tab for the ETL Developers group, add Tara, and explicitly grant the WriteMetadata permission to her.

Note: Don't assume that someone who has only indirect ☑ settings on someone else's **Authorization** tab has not been delegated management. The best way to check for delegation of an identity is to check each entry in the **Users and Groups** list box on that identity's **Authorization** tab to see whether there are any explicit ☑ or ACT ☑ (green) grants of the WriteMetadata permission.

Include a User in Batch Synchronization

Only users, groups, and roles that have an external identity can participate in batch synchronization. The external identity serves as a synchronization key.

1. Select the user, right-click, and access the user's **General** tab.
2. Click **External Identities**.
3. Click **New**. In the **Identifier** field, provide a value that identifies the user in an external source.

   CAUTION:

   An inaccurate external identity value can cause inadvertent deletion of an identity during the synchronization process. Make sure that any external identity value that you add corresponds to a key ID value in the tables that you extract from your primary user registry.

   Note: SAS doesn't enforce uniqueness when you store external identity values.

   Note: The synchronization process uses only the first external identity in each list.

Tips for Using the List of Identities

This list explains how you can modify the main display of users, groups, and roles:

- To sort the list of identities in ascending or descending order, click a column heading.
- To revert to the order in which identities were added to the repository, right-click a column heading and select **Sort Original**.
- To hide a column, right-click the column heading and select **Hide Column**.
- To show a hidden column, right-click any column heading and select **Show** ⇨ ***the column name***.
- To change the width of a column, click and drag the edge of the column heading.

- To move a column, click and drag the column heading.
- To set a different default view, select the **User Manager** node on the **Plug-ins** tab, right-click and select **Options**.
- To limit the type of identities displayed, clear the **Show Users**, **Show Groups** or **Show Roles** check boxes.
- To filter the list of identities displayed, select the **Search** radio button.

Note: The **User, Group, or Role** column lists display names. For an identity that doesn't have a display name, the name is listed instead.

To find an identity in the main display:

1. Make sure the correct repository is selected at the top of the **Plug-ins** tab.
2. Ensure that the appropriate **Show** check box is selected.
3. If you can't easily locate the identity, select the **Search** radio button, specify criteria, and click **Search Now**. The generated list includes all identities that meet all of the specified criteria. To specify additional criteria, click **Advanced**. When advanced search criteria are applied, a yellow symbol Ⓐ appears on the **Advanced** button.

Chapter 3
Access Management Tasks

Checking Permissions

Checking permissions is an item-centric activity. To view someone's permissions, do not begin by finding that person's user definition. Instead, begin by navigating to an item that you are interested in, opening that item's Properties dialog box, and selecting the item's **Authorization** tab.

How to Interpret the Authorization Tab

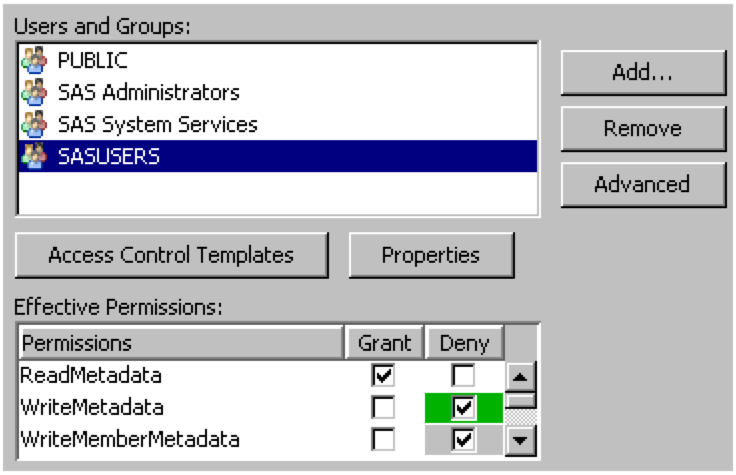
The List of Names

The **Users and Groups** list box includes only those users and groups who participate in the current item's settings. An identity participates if they are included in any of these places:

- the repository ACT's **Permission Pattern** tab
- a setting that this item inherits from a parent item
- an applied ACT's **Permission Pattern** tab
- an explicit setting on this item

Note: You can't remove identities that participate through the repository ACT, an applied ACT, or an inherited setting.

Any restricted user who is not listed has the access of their closest listed group. For each unlisted user, group memberships and identity precedence determine which listed group is closest. For example, the closest listed group for an administrator is usually SAS Administrators, and the closest listed group for a regular user is often SASUSERS.



Users and Groups:

- PUBLIC
- SAS Administrators
- SAS System Services
- SASUSERS

Effective Permissions:

Permissions	Grant	Deny
ReadMetadata	☑	□
WriteMetadata	□	☑
WriteMemberMetadata	□	☑

The List of Permissions

The **Effective Permissions** list box displays the metadata layer access that the selected user or group has for the current item. Effective permissions are a calculation of the net effect of all applicable permission settings. However, effective permissions don't reflect role-based constraints or access in other layers such as the operating system.

This table explains the significance of the check box colors:

Table 3.1 Significance of Color in the Permissions List

Color	Term	Significance
☑ (clear)*	Explicit	The permission is set on the current item and assigned to the selected identity.
☑ (green)	ACT	The permission comes from an applied ACT whose pattern explicitly assigns the grant or denial to the selected identity.

Color	Term	Significance
☑ (gray)	Indirect	The permission comes from someone else (the unrestricted role or a group that has an explicit or ACT setting) or somewhere else (a parent item or the repository ACT).[**]

[*] Explicit settings are usually white because the background color for the permissions list box is usually white.

[**] For the WriteMemberMetadata permission, gray means that the setting either mirrors the setting for the WriteMetadata permission or is derived from group settings.

How to Check the Permissions of an Unlisted User

Basic Technique

Click **Add** and temporarily add the user to the **Authorization** tab.

Note: Each restricted identity that you add gets an explicit ☑ grant of the ReadMetadata permission (unless the user has the unrestricted role). If you remove the user from the **Users and Groups** list box, the automatically created explicit grant of ReadMetadata is deleted.

Advanced Technique

If you are unrestricted, an **Advanced** button on each item's **Authorization** tab provides access to the item's **Explore Authorizations** tab. On the **Explore Authorizations** tab, you can add any user or group and view their permissions for the current item. You can't change settings on the **Explore Authorizations** tab. It is not necessary to remove identities from this tab. This tab is for investigation only.

Note: Both the **Authorization** tab and the advanced **Explore Authorizations** tab always display effective permissions.

Which Items are Parents to This Item?

If you are unrestricted, an **Advanced** button on each item's **Authorization** tab provides access to the **Inheritance** tab. On this tab, you can trace the current item's parents.

The **Inheritance** tab displays a tree of items, organized by their security relationships. The first item in the tree is always the current item. If the current item has an immediate parent other than the repository ACT, you can expand the first node in the tree to see those parents. You can continue expanding nodes to further trace the inheritance. The repository-level parent (the repository ACT 🔒) is not displayed in the tree.

TIP When you move from the **Folders** tab to the **Inheritance** tab, there is a shift in orientation. On the **Folders** tab, you expand parent nodes in order to get to an item that you are interested in. On the **Inheritance** tab, you begin with the item that you are interested in and expand nodes to move up that item's inheritance path.

These examples describe how the **Inheritance** tab displays inheritance paths:

- Each user, group, role, ACT, and application server inherits only from the repository ACT. On the **Inheritance** tab for any of these items, only the item itself is listed.
- Each BI content item (such as a report, information map, folder, or stored process) inherits from one immediate parent. On the **Inheritance** tab for each of these items, there is one expandable node immediately below the item.

Who Can Set Permissions?

Table 3.2 *Requirements for Setting Permissions*

Task	Requirements
Set permissions on an item	WriteMetadata for the item
Change the permission pattern on an ACT	WriteMetadata for the ACT
Designate a different repository ACT	WriteMetadata for the ACT

Note: In SAS Management Console, you can't see the **Authorization Manager** or any **Authorization** tabs unless you have the Authorization Manager capability.

Add an Explicit Grant or Denial

1. Navigate to the item that you want to protect or make available.
2. On the item's **Authorization** tab, select a user or group. Or, if you want to assign a permission to someone who is not listed, click **Add**. Each restricted identity that you add gets an explicit ☑ grant of the ReadMetadata permission.
3. In the **Effective Permissions** list box, select check boxes to adjust the settings for the currently selected identity. Each click adds an explicit setting to the item's protections (except that clicking an explicit ☑ setting removes that setting and reveals an underlying grant or denial).

   Note: If the identity that is selected in the **Users and Groups** list box has the unrestricted role, all permissions are granted and you can't change the settings.
4. Repeat steps 2 and 3 for any other identities whose access to this item you want to adjust.
5. Review the settings for each identity in the **Users and Groups** list box. This is important because settings that you add for a group can affect access for all members of that group. For example, a denial that you add for the PUBLIC group blocks access for all restricted users, unless there are other explicit ☑ or ACT ☑ (green) grants. You must offset a broad explicit denial with explicit or ACT grants for any restricted identities whose access you want to preserve.
6. In the Properties dialog box, click **OK** to save your changes.

TIP It is easy to set explicit grants and denials on each item that you want to protect or make available. However, managing a large number of individual permission settings can be cumbersome. See "Tips for Efficiently Using Permissions" on page 38.

Use an Access Control Template (ACT)

Why Use ACTs?

Use ACTs to avoid having to repeatedly set the same explicit permissions for the same identities on multiple items. When you apply an ACT to an item, the ACT settings are added to the item's protections.

How to Use an ACT

1. Determine whether there is an existing ACT that you can use.

   a. On the **Plug-ins** tab of SAS Management Console, select **Authorization Manager ⇨ Access Control Templates**.

   b. On the **Permission Pattern** tab of each ACT, examine the settings for each identity. If you don't find an appropriate ACT, consider using a combination of ACTs and explicit settings or creating a new ACT.

   Note: Don't confuse an ACT's **Authorization** tab with its **Permission Pattern** tab. Settings on an ACT's **Authorization** tab affect who can access that ACT; settings on an ACT's **Permission Pattern** tab affect access to the items to which that ACT is applied.

2. When you have identified an ACT that you want to use, navigate to an item to which you will add that ACT's settings. On the item's **Authorization** tab, click **Access Control Templates**.

3. Expand the nodes in the **Available** list box, move the ACT to the **Currently Using** list box, and click **OK**.

   Note: The repository ACT 🔒 is typically not in the **Currently Using** list box because that ACT is typically not applied to any items.

   Note: You can apply multiple ACTs. For example, on a report folder, you might apply one ACT that grants read access to a SALES group and also apply another ACT that grants read and write access to a Report Creators group.

4. On the item's **Authorization** tab, notice that the **Users and Groups** list box now includes the identities that participate in the ACT that you selected. Select each identity and verify that the revised settings are as you expect. On the **Authorization** tab of an item to which an ACT is applied, settings that are explicit ☑ in the ACT's pattern are green ✅.

5. In the item's Properties dialog box, click **OK**.

Create a Custom ACT

Why Create Custom ACTs?

Several predefined ACTs are provided. To further centralize access management, create an ACT for each access pattern that you use multiple times. This list outlines common patterns and provides tips:

- It is often useful to create ACTs to manage read access for different business units.
- It is often useful to create an ACT that manages write access for a functional group that includes users from multiple business units.
- You don't have to capture all of an item's protections in one ACT. You can use combinations of ACTs, explicit settings, and inherited settings to define access to an item.

How to Create a Custom ACT

1. Review the existing ACTs to make sure that the pattern doesn't already exist.

   a. On the **Plug-ins** tab of SAS Management Console, select **Environment Management ⇨ Authorization Manager ⇨ Access Control Templates**

   b. On the **Permission Pattern** tab of each ACT, examine the settings for each identity.

   Note: Don't confuse an ACT's **Authorization** tab with its **Permission Pattern** tab. Settings on an ACT's **Authorization** tab affect who can access that ACT; settings on an ACT's **Permission Pattern** tab affect access to the items to which that ACT is applied.

2. Create the ACT.

   a. On the **Plug-ins** tab in SAS Management Console, select **Authorization Manager ⇨ Access Control Templates**.

   b. Right-click and select **New Access Control Template**.

   c. On the **General** tab, enter a name. It is a good idea to use the description field to document the intended purpose of the ACT.

   d. On the **Permission Pattern** tab, add one or more identities and select check boxes. Each restricted identity that you add gets a grant of the ReadMetadata permission in the pattern.

   Note: The pattern is a collection of settings that will be added to the protections for each item to which you apply this ACT. Any gray check boxes come from group memberships. The gray settings aren't part of the ACT's pattern; they just show the net effect of that pattern for the selected identity.

   Note: For each identity, the pattern can provide a grant, a deny, or a blank setting for each permission. Settings that are unspecified (neither granted nor denied) in an ACT's pattern have no effect when that ACT is applied to an item.

   Note: If the identity that is selected in the **Users and Groups** list box has the unrestricted role, all permissions are granted and you can't change the settings.

   e. On the **Authorization** tab, define who can do what to the new ACT. It is important to prevent regular users from modifying or removing an ACT. A typical approach is to add an explicit ☑ denial of WriteMetadata for PUBLIC and an offsetting explicit grant of WriteMetadata for SAS Administrators.

f. In the Properties dialog box, click **OK**. The new ACT is now in the list of ACTs under **Authorization Manager** ⇨ **Access Control Templates**.

3. Apply the ACT to one or more items. For each item to which you want to add the ACT's settings, complete these steps:

   a. Navigate to the item's **Authorization** tab.

   b. Click **Access Control Templates**.

   c. In the **Available** list box, open the nodes and move the new ACT to the **Currently Using** list box. Click **OK** to close the dialog box.

   d. On the item's **Authorization** tab, verify that the revised settings are as you expect. On the **Authorization** tab of an item to which an ACT is applied, settings that are explicit ☑ in the ACT's pattern are green ✅.

   Note: The applied ACT contributes its settings to the item's protections. The item can also have explicit settings and other applied ACTs (as well as inherited settings).

4. If necessary, adjust the ACT's pattern. The advantage of using an ACT is that you can change the pattern without revisiting the items to which the pattern is applied. Simply make changes on the ACT's **Permission Pattern** tab.

Update or Delete an ACT

CAUTION:

One ACT can protect thousands of items. Changes that you make to an ACT's pattern affect every item that ACT is applied to.

CAUTION:

When you delete an ACT, you lose all of that ACT's associations to items where it is applied. Creating a new ACT with the same name does not restore those associations.

1. On the **Plug-ins** tab of SAS Management Console, navigate to **Authorization Manager** ⇨ **Access Control Templates** and select an ACT.

2. To modify the ACT's pattern:

   a. Right-click and select **Properties**.

   b. Adjust settings on the **Permission Pattern** tab.

   Note: Don't confuse an ACT's **Authorization** tab with its **Permission Pattern** tab. Settings on an ACT's **Authorization** tab affect who can access that ACT; settings on an ACT's **Permission Pattern** tab affect access to the items to which that ACT is applied.

   c. Click **OK** to save your changes.

   d. (Optional) Navigate to the Properties dialog box of an item that uses this ACT and verify that the revised settings are as you expect. On the **Authorization** tab of an item to which an ACT is applied, settings that are explicit ☑ in the ACT's pattern are green ✅.

3. To delete the ACT, right-click and select **Delete**. In the confirmation message box, click **Yes**.

Set a Permission Condition

Permission conditions limit explicit grants of the Read permission so that different users access different subsets of data.

1. Access the **Authorization** tab of the dimension for which you are defining a permission condition.

   Note: In SAS Management Console, access dimensions on the **Plug-ins** tab under **Authorization Manager** ⇨ **Resource Management** ⇨ **By Location** ⇨ **<server>** ⇨ **<OLAP schema>** ⇨ **cube**.

2. Select or add the identity whose access to measures you want to limit.
3. In the permissions list, add an explicit ☑ grant of the Read permission for the selected identity.
4. Click the **Add Authorization** button.

   Note: If the **Edit Authorization** button is displayed, a condition already exists for the selected user or group.

5. In the Add Authorization dialog box, select the **Create an advanced MDX expression** radio button and click **Build Formula**.
6. In the Build Formula dialog box, define a condition that filters the data as appropriate for the selected identity. For detailed assistance, click the **Help** button in the dialog box.

Note: You can manage permission conditions for OLAP dimensions from SAS OLAP Cube Studio and SAS Data Integration Studio as well as SAS Management Console.

Note: To define and manage permission conditions for information maps, use SAS Information Map Studio.

See Also

"Fine-Grained Controls" on page 9

Adjust the Repository-Level Settings

Why Adjust the Repository-Level Settings?

CAUTION:

Altering the repository-level settings for service identities can prevent necessary access. We recommend that you do not change these settings.

This list highlights key points about working with repository-level settings for a foundation repository:

- If you want some or all users to have default read access to all data, grant the Read permission at the repository level.
- If you want to experiment with narrowing repository-level access, we recommend that you create a new ACT and designate that ACT as the repository ACT (instead of modifying the original repository ACT).
- Typically, all users need ReadMetadata and WriteMetadata access to the foundation repository. It is appropriate for the SASUSERS group to have these permissions on the repository ACT's **Permission Pattern** tab.

Make Changes to the Repository ACT

To make changes to the repository ACT:

1. On the **Plug-ins** tab in SAS Management Console, select **Authorization Manager** ⇨ **Access Control Templates**.
2. In the display area, select the repository ACT .
3. Right-click and select **Properties**. Make changes on the **Permission Pattern** tab. Each restricted identity that you add gets a grant of the ReadMetadata permission in the pattern.

   For example, to give all registered users default read access to all data, select the SASUSERS group and then select the **Grant** check box for the Read permission.

   Note: Any gray check boxes are settings that come from the selected identity's group memberships.

   Note: Don't confuse the **Permission Pattern** tab with the **Authorization** tab. Settings on the **Authorization** tab affect who can access this ACT; settings on this **Permission Pattern** tab define access to the repository.

   Note: There is no reason to specify grants or denials of the WriteMemberMetadata permission as part of the repository-level settings. Unlike other permissions, the WriteMemberMetadata permission is never inherited from one item to another.

   Note: In the repository ACT's pattern, an identity that has a blank setting for a particular permission (neither a grant nor a denial) is denied that permission.

Designate a Different ACT to Serve as the Repository ACT

To designate a different repository ACT:

1. Identify or create an ACT that has the repository-level settings that you want to use.
2. On the **Plug-ins** tab in SAS Management Console, under **Authorization Manager** ⇨ **Access Control Templates**, select the ACT that you want to use to define repository-level access.
3. Right-click and select **Repository ACT**. In the confirmation message box, click **Yes**.

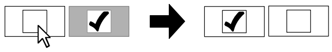
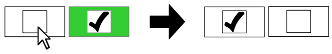
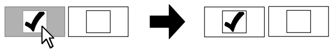
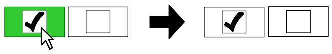
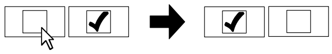
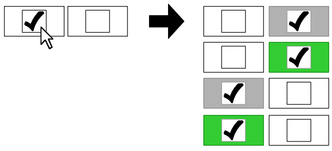
   In the list of ACTs under **Authorization Manager** ⇨ **Access Control Templates**, the repository ACT icon is now displayed next to the newly designated repository ACT. The ACT that originally served as the repository ACT still exists, but it is no longer in use.

Note: To revert to the original repository ACT, select that ACT and repeat step 3.

What Happens When I Select a Check Box?

The following table explains what happens when you select a check box on the **Authorization** tab. Each pair of check boxes depicts the grant and denial settings for a permission in the **Effective Permissions** list. In each row, the pointer ↖ indicates an action (a mouse click) that occurs between the before and the after.

Table 3.3 Mechanics of the Effective Permissions List

Before and After	Explanation
	A new explicit setting overrides and hides the opposing indirect (gray) setting.
	A new explicit setting overrides and hides the opposing ACT (green) setting.
	A new explicit setting is added on top of the matching indirect (gray) setting.
	A new explicit setting is added on top of the matching ACT (green) setting.
	A new explicit setting replaces the opposing explicit setting.
	The explicit setting is removed and one of these underlying indirect (gray) or ACT (green) settings is revealed.

Tips for Efficiently Using Permissions

Assign Permissions To Groups

You can simplify access control management by assigning permissions to groups rather than to individual users. These examples assume that there are not other explicit or ACT settings on the item:

- To allow only unrestricted users to access an item, set denials on that item for the PUBLIC group.
- To enable only registered users to access an item, set denials for the PUBLIC group and then grant access back to the SASUSERS group.
- To enable only ETL developers and unrestricted users to access an item, create a group for the ETL developers. Then deny permissions to the PUBLIC group and grant access back to the ETL developers group.

Use Folders To Organize Content

You can simplify access control management by creating a folder structure that reflects the access distinctions that you want to make. Instead of setting permissions on each individual item, set permissions on the folders. The items in a folder inherit the folder's effective permissions.

TIP To protect the folder structure, do not grant WriteMetadata permission on a folder to someone for whom WriteMemberMetadata permission is sufficient.

Centralize Permissions with ACTs

You can simplify access control management by using ACTs. An ACT is a reusable named pattern of settings that you can apply to multiple items. Each ACT consists of these elements:

- a list of users and groups
- an indication of whether each permission is granted, denied, or unspecified for each user and group in the list

Deny Broadly, Grant Selectively (To the Extent Possible)

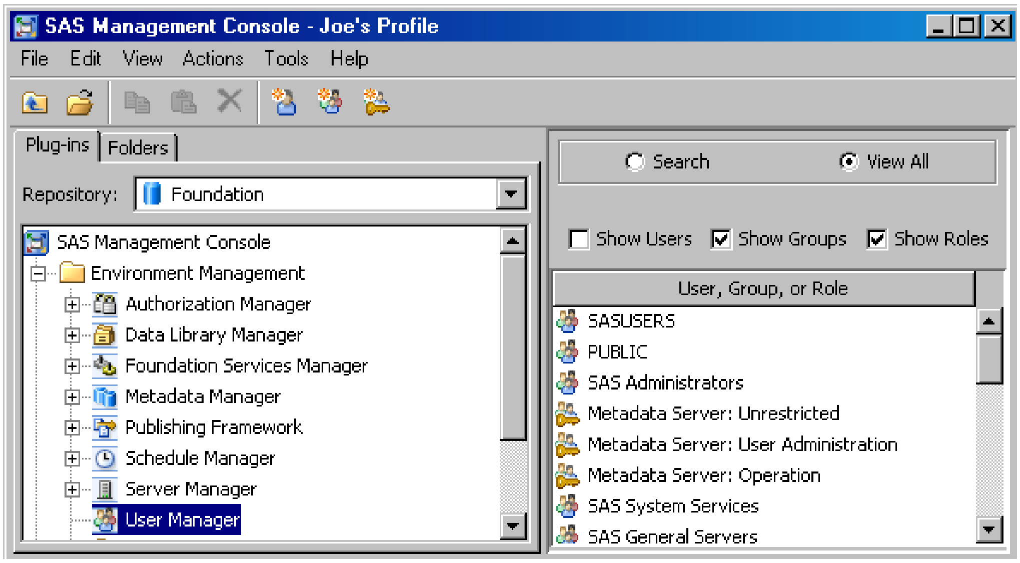
Assign denials to the broadest group (PUBLIC) and then add offsetting grants for users or groups whose access you want to preserve. Deny access at the highest point of control and then grant access back on specific containers or items. These constraints apply:

- The highest point of control is the repository-level settings that are defined on the foundation repository ACT's **Permission Pattern** tab. The security model requires that participating users have ReadMetadata and WriteMetadata access at this level, so broadly denying access here is not a workable approach. Instead, use the next point of control, which is the top of the folder tree on the **Folders** tab.
- Within the folder tree, users need a clear path of grants of ReadMetadata in order to navigate to the items that they use. For this permission, setting denials on folders at a high level is not a workable approach.

Appendix 1
Visual Reference

User Administration: Selected Screenshots

The Location for Managing Users, Groups, and Roles



Repository: Foundation

- SAS Management Console
  - Environment Management
    - Authorization Manager
    - Data Library Manager
    - Foundation Services Manager
    - Metadata Manager
    - Publishing Framework
    - Schedule Manager
    - Server Manager
    - User Manager

○ Search ● View All

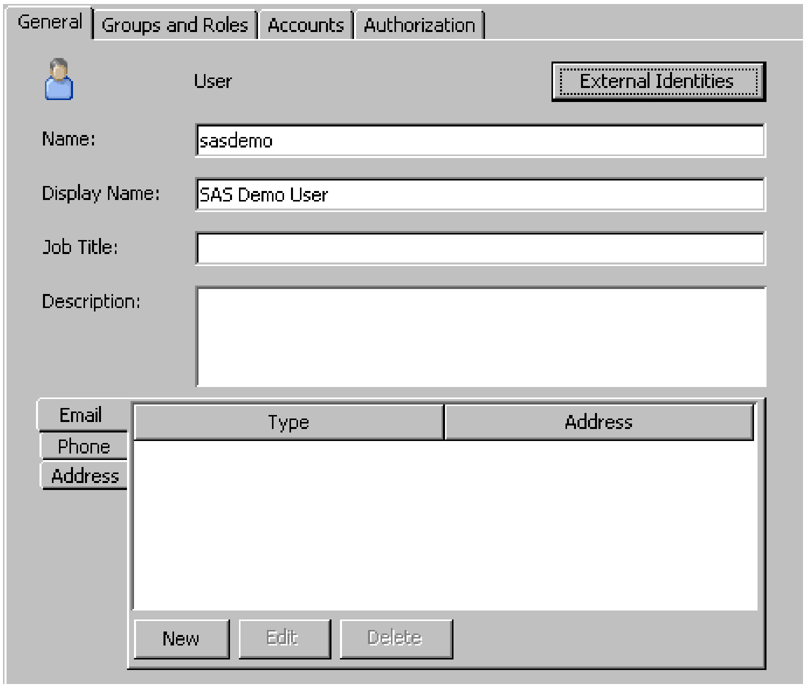
☐ Show Users ☑ Show Groups ☑ Show Roles

User, Group, or Role

SASUSERS
PUBLIC
SAS Administrators
Metadata Server: Unrestricted
Metadata Server: User Administration
Metadata Server: Operation
SAS System Services
SAS General Servers

User Properties

***Display A1.1** User: General*



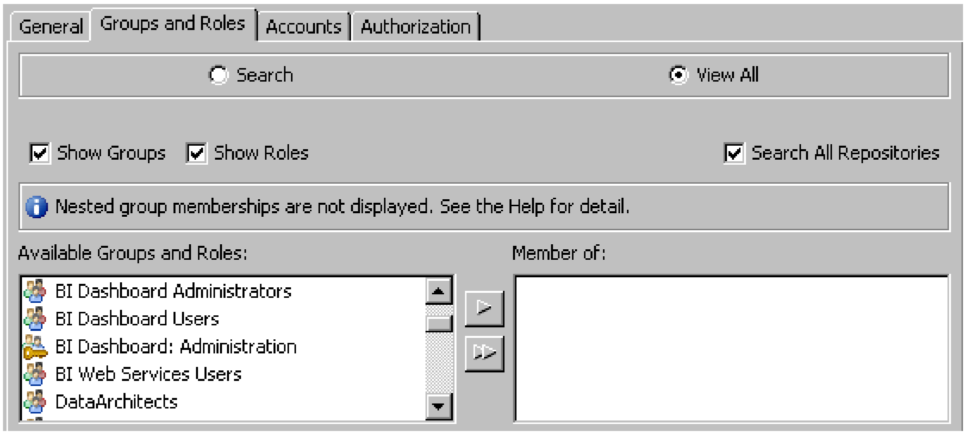
Name: sasdemo

Display Name: SAS Demo User

Job Title:

Description:

***Display A1.2** User: Groups and Roles*



○ Search ● View All

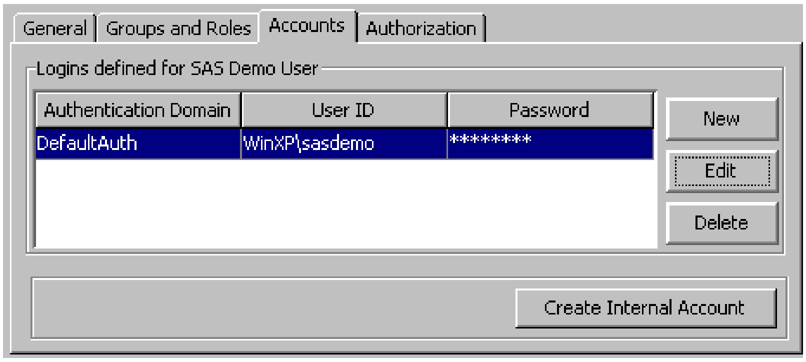
☑ Show Groups ☑ Show Roles ☑ Search All Repositories

Nested group memberships are not displayed. See the Help for detail.

Available Groups and Roles:

- BI Dashboard Administrators
- BI Dashboard Users
- BI Dashboard: Administration
- BI Web Services Users
- DataArchitects

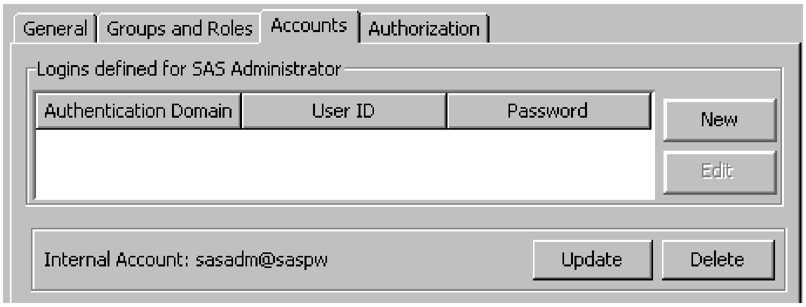
Display A1.3 *User: Accounts (with a login)*



Logins defined for SAS Demo User

Authentication Domain	User ID	Password
DefaultAuth	WinXP\sasdemo	********

Display A1.4 *User: Accounts (with an internal account)*

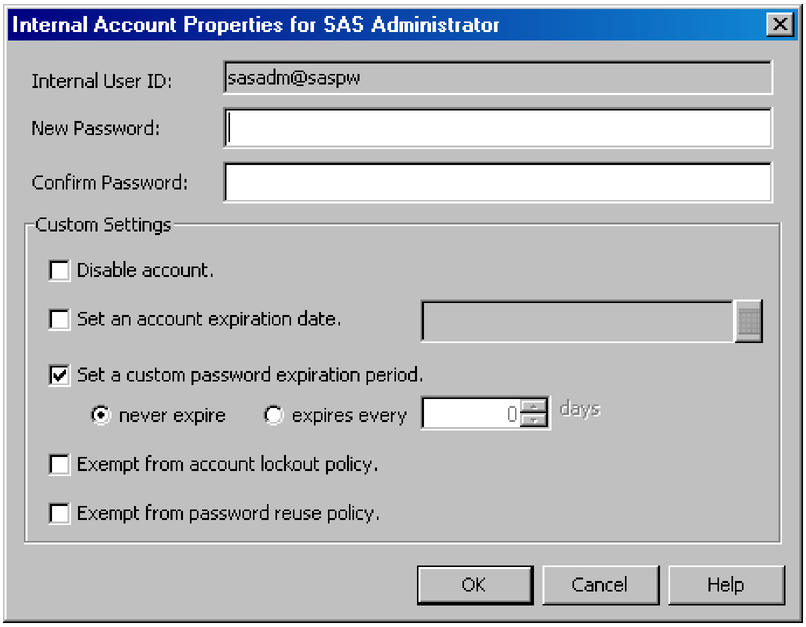


Logins defined for SAS Administrator

Authentication Domain	User ID	Password

Internal Account: sasadm@saspw

Display A1.5 *Internal Account Properties*



Internal Account Properties for SAS Administrator

Internal User ID: sasadm@saspw

New Password:

Confirm Password:

Custom Settings

□ Disable account.

□ Set an account expiration date.

☑ Set a custom password expiration period.

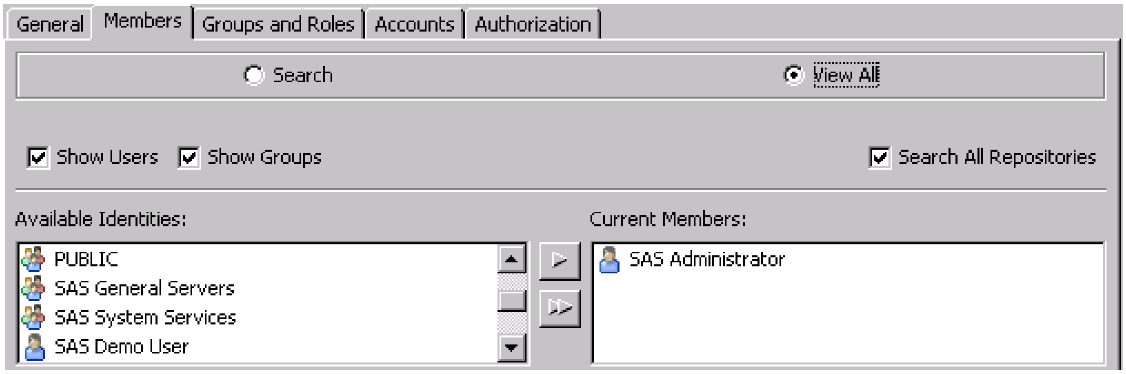
● never expire ○ expires every 0 days

□ Exempt from account lockout policy.

□ Exempt from password reuse policy.

Group Properties

Display A1.6 *Group: Members*



○ Search ● View All

☑ Show Users ☑ Show Groups ☑ Search All Repositories

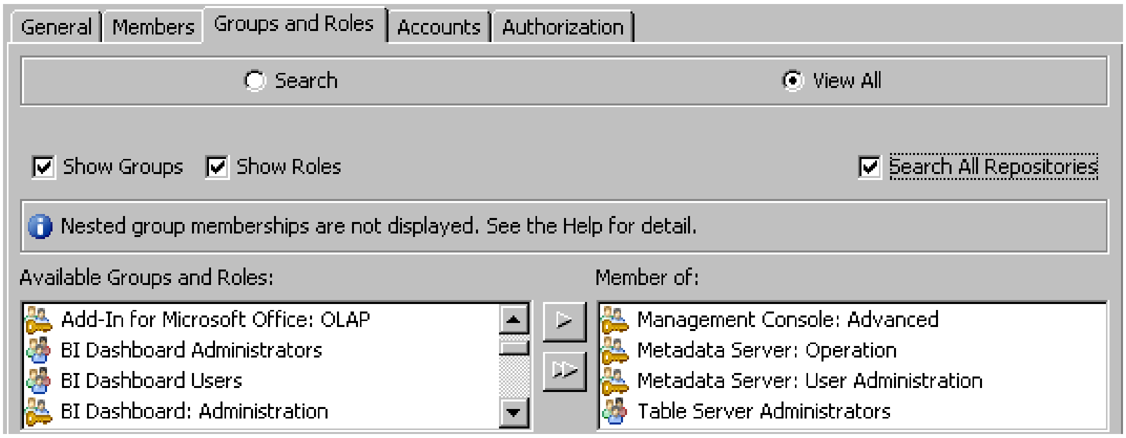
Available Identities:

- PUBLIC
- SAS General Servers
- SAS System Services
- SAS Demo User

Current Members:

- SAS Administrator

Display A1.7 *Group: Groups and Roles*



○ Search ● View All

☑ Show Groups ☑ Show Roles ☑ Search All Repositories

Nested group memberships are not displayed. See the Help for detail.

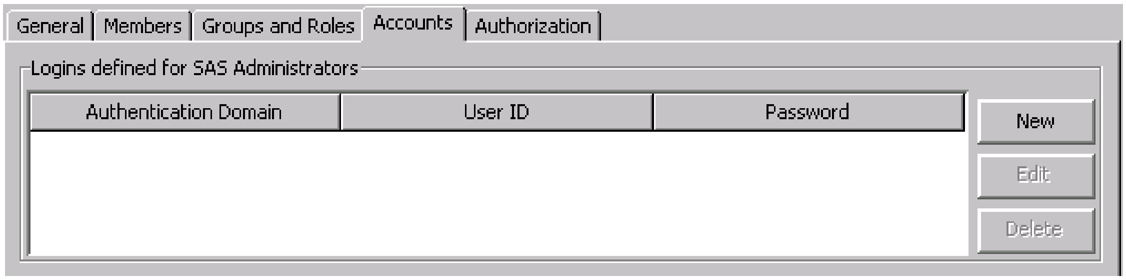
Available Groups and Roles:

- Add-In for Microsoft Office: OLAP
- BI Dashboard Administrators
- BI Dashboard Users
- BI Dashboard: Administration

Member of:

- Management Console: Advanced
- Metadata Server: Operation
- Metadata Server: User Administration
- Table Server Administrators

Display A1.8 *Group: Accounts*

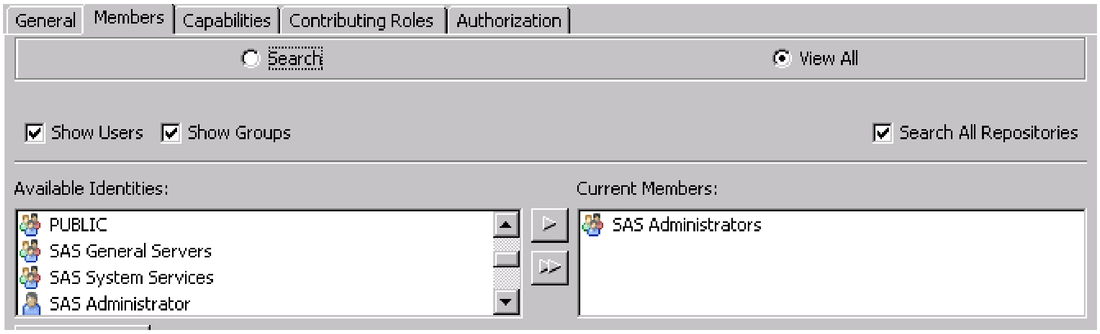


Logins defined for SAS Administrators

Authentication Domain	User ID	Password

Role Properties

Display A1.9 *Role: Members*



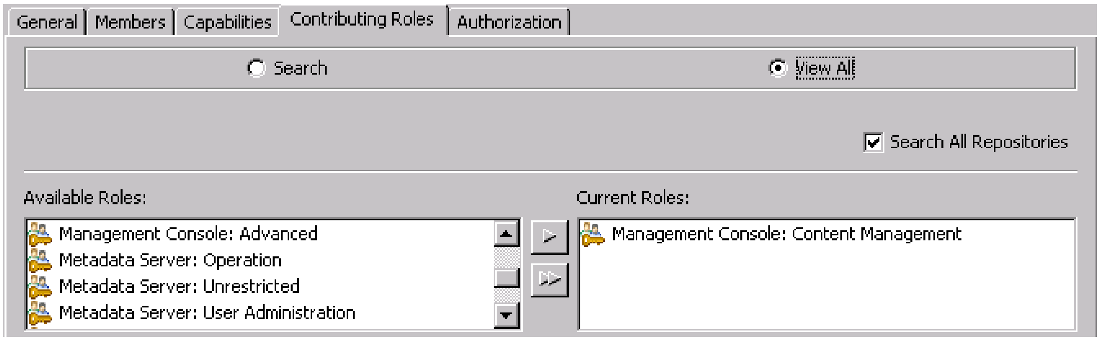
○ Search ● View All

☑ Show Users ☑ Show Groups ☑ Search All Repositories

Available Identities:
- PUBLIC
- SAS General Servers
- SAS System Services
- SAS Administrator

Current Members:
- SAS Administrators

Display A1.10 *Role: Contributing Roles (these settings are customized)*



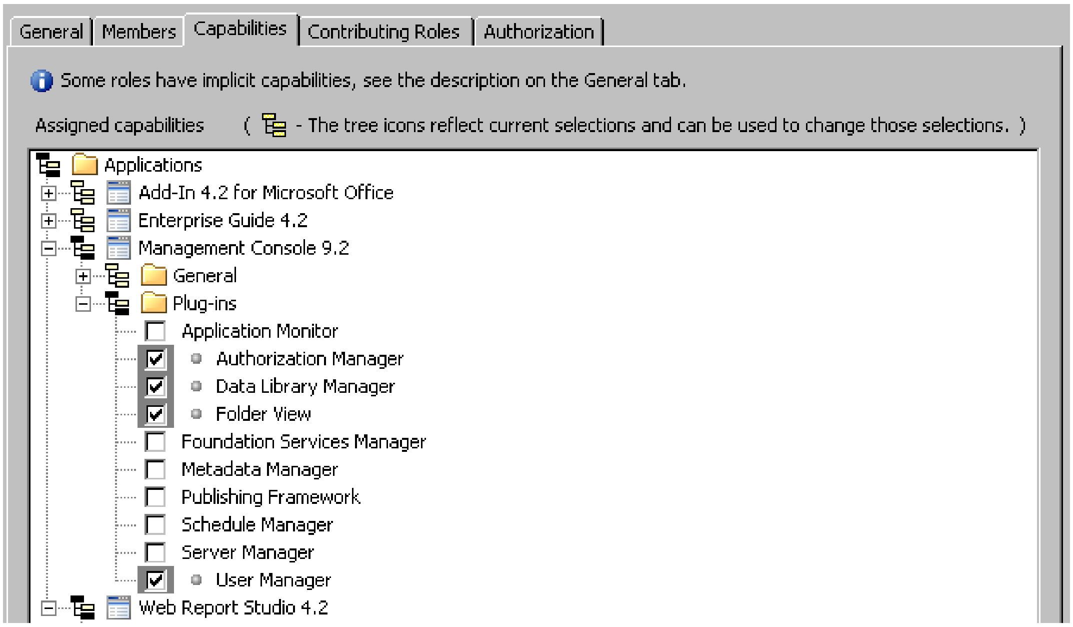
○ Search ● View All

☑ Search All Repositories

Available Roles:
- Management Console: Advanced
- Metadata Server: Operation
- Metadata Server: Unrestricted
- Metadata Server: User Administration

Current Roles:
- Management Console: Content Management

Display A1.11 *Role: Capabilities (these settings are customized)*



Some roles have implicit capabilities, see the description on the General tab.

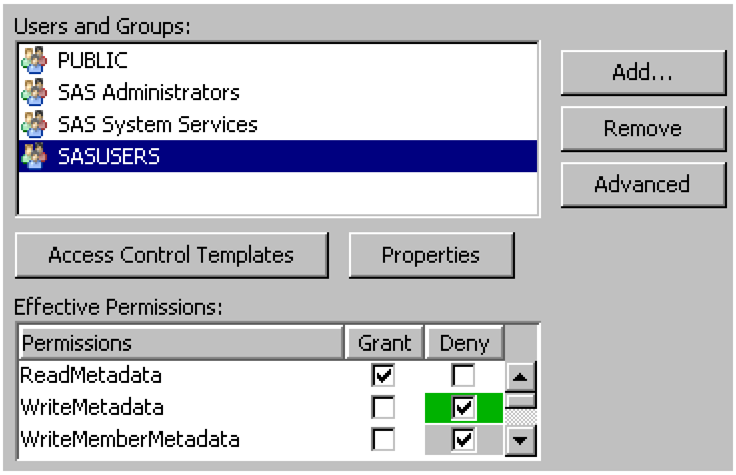
Assigned capabilities (- The tree icons reflect current selections and can be used to change those selections.)

- Applications
  - Add-In 4.2 for Microsoft Office
  - Enterprise Guide 4.2
  - Management Console 9.2
    - General
    - Plug-ins
      - □ Application Monitor
      - ☑ Authorization Manager
      - ☑ Data Library Manager
      - ☑ Folder View
      - □ Foundation Services Manager
      - □ Metadata Manager
      - □ Publishing Framework
      - □ Schedule Manager
      - □ Server Manager
      - ☑ User Manager
  - Web Report Studio 4.2

Access Management: Selected Screenshots

Authorization Interface

Display A1.12 *Authorization (for setting permissions)*



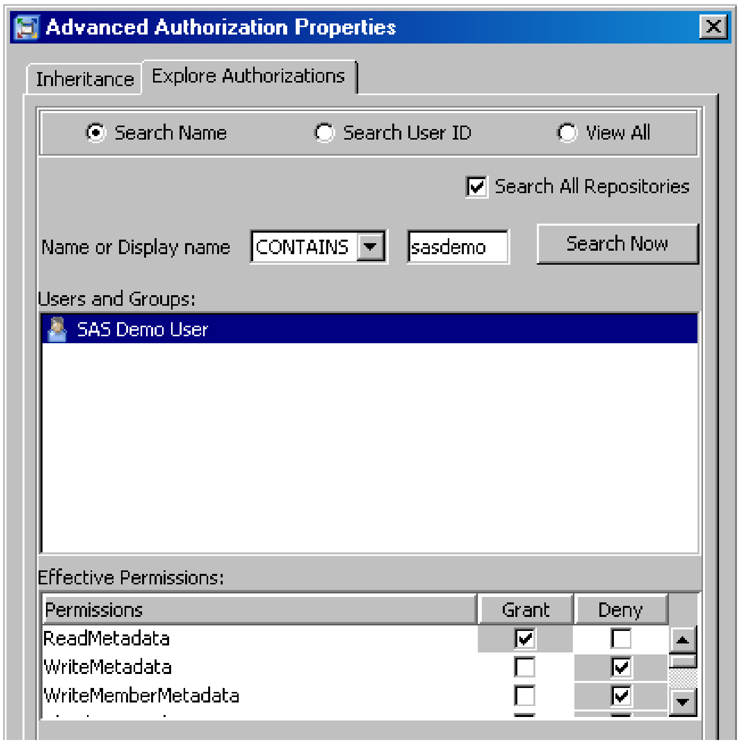
Users and Groups:

- PUBLIC
- SAS Administrators
- SAS System Services
- SASUSERS

Effective Permissions:

Permissions	Grant	Deny
ReadMetadata	☑	□
WriteMetadata	□	☑
WriteMemberMetadata	□	☑

Display A1.13 *Advanced: Explore Authorizations (for looking up permissions)*



● Search Name ○ Search User ID ○ View All

☑ Search All Repositories

Name or Display name CONTAINS sasdemo

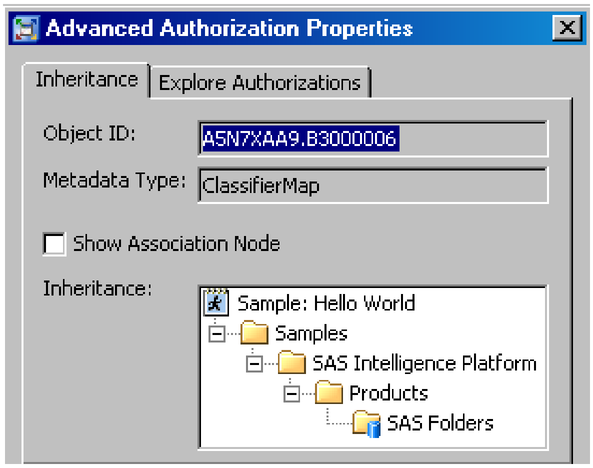
Users and Groups:

- SAS Demo User

Effective Permissions:

Permissions	Grant	Deny
ReadMetadata	☑	□
WriteMetadata	□	☑
WriteMemberMetadata	□	☑

Display A1.14 *Advanced: Inheritance for a Stored Process*



Advanced Authorization Properties

Object ID: A5N7XAA9.B3000006

Metadata Type: ClassifierMap

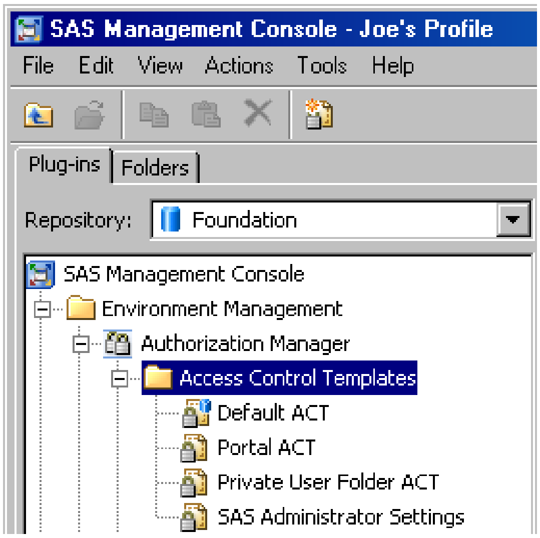
□ Show Association Node

Inheritance:

- Sample: Hello World
- Samples
  - SAS Intelligence Platform
    - Products
      - SAS Folders

Access Control Templates

Display A1.15 *The Location for Managing ACTs*

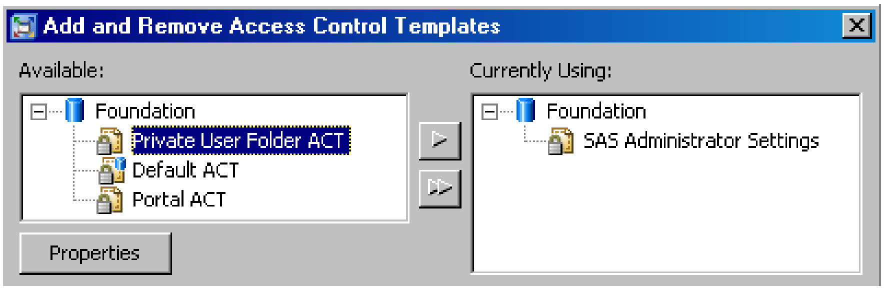


SAS Management Console - Joe's Profile

Repository: Foundation

- SAS Management Console
  - Environment Management
    - Authorization Manager
      - Access Control Templates
        - Default ACT
        - Portal ACT
        - Private User Folder ACT
        - SAS Administrator Settings

Display A1.16 *The List of ACTs That Are Applied to a Particular Item*



Add and Remove Access Control Templates

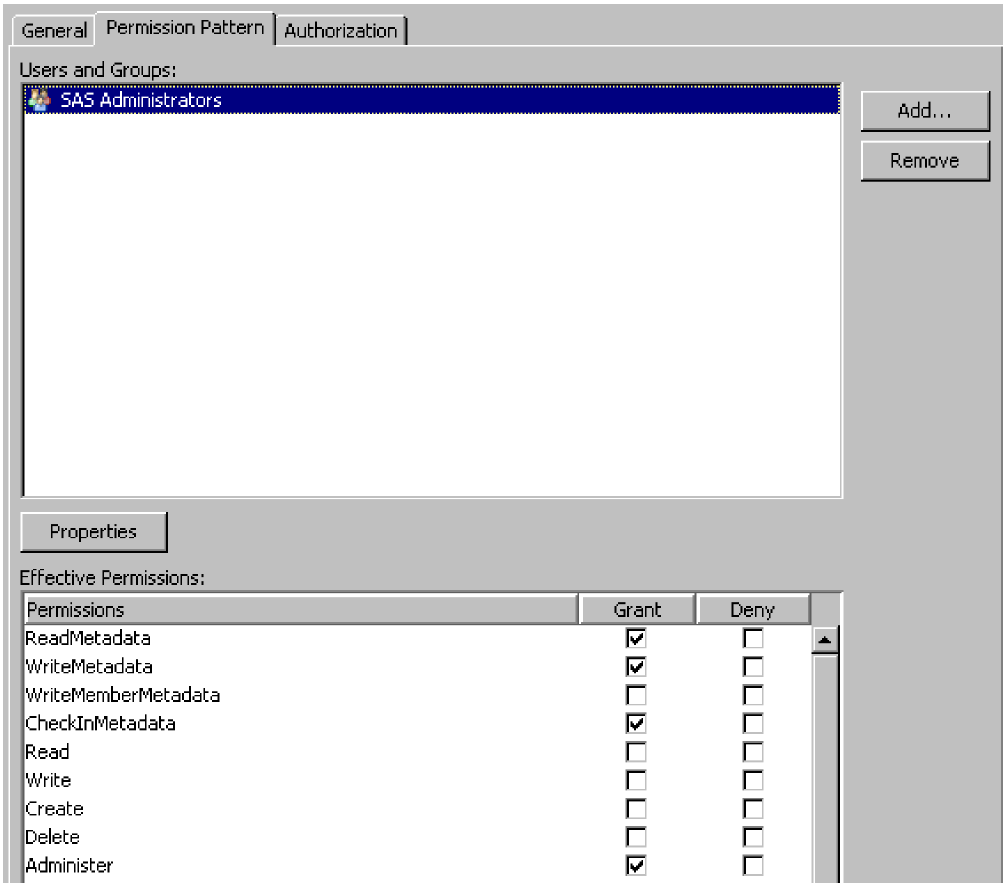
Available:

- Foundation
  - Private User Folder ACT
  - Default ACT
  - Portal ACT

Currently Using:

- Foundation
  - SAS Administrator Settings

Display A1.17 *The Pattern of Settings That a Particular ACT Provides*



Users and Groups:

SAS Administrators

Effective Permissions:

Permissions	Grant	Deny
ReadMetadata	☑	□
WriteMetadata	☑	□
WriteMemberMetadata	□	□
CheckInMetadata	☑	□
Read	□	□
Write	□	□
Create	□	□
Delete	□	□
Administer	☑	□

Glossary

access control template
a reusable named authorization pattern that you can apply to multiple resources. An access control template consists of a list of users and groups and indicates, for each user or group, whether permissions are granted or denied. Short form: ACT.

authentication
the process of verifying the identity of a person or process within the guidelines of a specific authorization policy.

authentication domain
a SAS internal category that pairs logins with the servers for which they are valid. For example, an Oracle server and the SAS copies of Oracle credentials might all be classified as belonging to an OracleAuth authentication domain.

authentication provider
a software component that is used for identifying and authenticating users. For example, an LDAP server or the host operating system can provide authentication.

authorization
the process of determining which users have which permissions for which resources. The outcome of the authorization process is an authorization decision that either permits or denies a specific action on a specific resource, based on the requesting user's identity and group memberships.

capability
an application feature that is under role-based management. Typically, a capability corresponds to a menu item or button. For example, a Report Creation capability might correspond to a New Report menu item in a reporting application. Capabilities are assigned to roles.

credentials
the user ID and password for an account that exists in some authentication provider.

external identity
a synchronization key for a user, group, or role. For example, employee IDs are often used as external identities for users. This is an optional attribute that is needed only for identities that you batch update using the user import macros.

identity
a user, group, or role definition.

internal account
a SAS account that you can create as part of a user definition. Internal accounts are intended for metadata administrators and some service identities; these accounts are not intended for regular users.

internal authentication
a process in which the metadata server verifies a SAS internal account. Internal authentication is intended for only metadata administrators and some service identities.

login
a SAS copy of information about an external account. Each login includes a user ID and belongs to one SAS user or group. Most logins do not include a password.

permission condition
a control that defines access to data at a low level, specifying who can access particular rows within a table or particular members within an OLAP cube. Such controls are typically used to subset data by a user characteristic such as employee ID or organizational unit. For example, an OLAP cube that contains employee information might have member-level controls that enable each manager to see the salary history of only that manager's employees. Similarly, a table that contains patient medical information might have row-level controls that enable each doctor to see only those rows that contain data about that doctor's patients.

restricted identity
a user or group that is subject to capability requirements and permission denials in the metadata environment. Anyone who isn't in the META: Unrestricted Users Role and isn't listed in the adminUsers.txt file with a preceding asterisk is a restricted identity.

role
a set of capabilities. In some applications, certain actions are available only to users or groups that have a particular role.

service identity
an identity or account that exists only for the purpose of supporting certain system activities and does not correspond to a real person. For example, the SAS Trusted User is a service identity.

unrestricted identity
a user or group that has all capabilities and permissions in the metadata environment due to membership in the META: Unrestricted Users Role (or listing in the adminUsers.txt file with a preceding asterisk).

Web authentication
a configuration in which users of Web applications are verified at the Web perimeter and the metadata server trusts that verification.

well-formed user definition
a user definition that includes a login with an appropriate user ID. For a Windows account, the user ID in the login must be qualified (for example, WIN\marcel or marcel@company.com). The login does not have to include a password. For metadata administrators and some service identities, it is appropriate to use an internal account instead of a login.

Index

Your Turn

We welcome your feedback.

- If you have comments about this book, please send them to **yourturn@sas.com**. Include the full title and page numbers (if applicable).
- If you have comments about the software, please send them to **suggest@sas.com**.

SAS® Publishing Delivers!

Whether you are new to the work force or an experienced professional, you need to distinguish yourself in this rapidly changing and competitive job market. SAS® Publishing provides you with a wide range of resources to help you set yourself apart. Visit us online at support.sas.com/bookstore.

SAS® Press

Need to learn the basics? Struggling with a programming problem? You'll find the expert answers that you need in example-rich books from SAS Press. Written by experienced SAS professionals from around the world, SAS Press books deliver real-world insights on a broad range of topics for all skill levels.

support.sas.com/saspress

SAS® Documentation

To successfully implement applications using SAS software, companies in every industry and on every continent all turn to the one source for accurate, timely, and reliable information: SAS documentation. We currently produce the following types of reference documentation to improve your work experience:

- Online help that is built into the software.
- Tutorials that are integrated into the product.
- Reference documentation delivered in HTML and PDF – **free** on the Web.
- Hard-copy books.

support.sas.com/publishing

SAS® Publishing News

Subscribe to SAS Publishing News to receive up-to-date information about all new SAS titles, author podcasts, and new Web site features via e-mail. Complete instructions on how to subscribe, as well as access to past issues, are available at our Web site.

support.sas.com/spn

sas® THE POWER TO KNOW®