SAS® Viya® 3.4 Administration: General Authorization

General Authorization: Overview ........................................ 2

General Authorization: How to (Authorization Window) ........................................ 2
  Introduction ........................................................................ 2
  Navigation ........................................................................ 3
  Examine Access .................................................................. 3
  Provide Access .................................................................... 4
  Limit Access ........................................................................ 4
  Add a Condition ................................................................... 4
  Edit a Condition ................................................................... 5
  Delete a Condition ............................................................... 5
  Remove a Direct Setting ......................................................... 5
  Identify the Source of Effective Access ................................... 6

General Authorization: How to (Rules Page) ........................................ 6
  Introduction ........................................................................ 6
  Navigation ........................................................................ 6
  Rules Page .......................................................................... 6
  Add a Rule .......................................................................... 7
  Edit a Rule .......................................................................... 7
  Copy a Rule .......................................................................... 8
  Delete a Rule ....................................................................... 8
  Edit a Condition ................................................................... 8
  Delete a Condition ............................................................... 9
  Locate a Particular Rule ......................................................... 9
  Extend the Ability to Create Top-Level Folders ......................... 10

General Authorization: Concepts ............................................ 10
  Scope ................................................................................ 10
  Key Terms .......................................................................... 10
  Principals ........................................................................... 11
  Administrators ..................................................................... 11
  Inheritance .......................................................................... 11
  Permissions .......................................................................... 12
  Permissions by Task ............................................................ 13
  Rule Targets ......................................................................... 15
General Authorization: Overview

To learn about the general authorization system, see Concepts.

To manage access, use the interface that best meets your needs. Here are suggestions:

- To adjust access to content (such as folders and reports), use the Authorization window.
- To manipulate rules directly, use the Rules page or the command-line interface.

General Authorization: How to (Authorization Window)

Introduction

These instructions explain how to set permissions on content (such as folders and reports) using the Authorization window.
Navigation

To access the Authorization window from SAS Environment Manager:

1. In the applications menu ( ), under Administration, select Manage Environment.
2. In the vertical navigation bar, click .
3. Locate and select the object.
4. Right-click, and select View authorization or Edit authorization.

Note: If the Edit authorization item is not available, you are not authorized to modify access to the object.

**TIP** The Authorization window is also available in SAS Drive to users who assume their membership in the SAS Administrators group. To expand or reduce that access, see Access to Functionality. (That access is provided by the Read permission on the following object URI: /authorizationDialog.)

Examine Access

For each principal and permission, the following icons describe effective (net) access to the current object:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔</td>
<td>Authorized</td>
</tr>
<tr>
<td>○</td>
<td>Conditional</td>
</tr>
<tr>
<td>☞</td>
<td>Not Authorized</td>
</tr>
<tr>
<td>◄</td>
<td>Unknown</td>
</tr>
<tr>
<td>◈</td>
<td>Direct (indicates that effective access comes from a direct setting)*</td>
</tr>
<tr>
<td>◅</td>
<td>Not Authorized (but can share)**</td>
</tr>
</tbody>
</table>

* This icon indicates that effective access comes from a permission that is directly assigned to the specified principal on the current object. If a direct setting exists but does not win (does not determine effective access), a diamond is not displayed.

** This icon is applicable to only the Secure and Secure (convey) columns. This icon is displayed only if Secure access is not granted and sharing is possible. If Secure access is authorized, only the Authorized icon is displayed, because the ability to share is inherent in Secure access.

The scope of the display is as follows:

- There is always a row for Authenticated Users.
- There is always a row for you, the currently connected user who is using the display.
- There is a row for each principal that is assigned to a rule that affects access to the current object. The exception is that internal service principals (for example, sasapp or sas.folders) are not displayed in the Authorization window.
- If you add an identity and do not give that identity at least one direct setting, that identity is automatically removed from the display.
You cannot directly remove a row. If you remove all direct settings for an identity and there is no other reason for that identity to be displayed, the identity is automatically removed from the display.

For a non-container object (such as a report), only the Read, Update, Delete, and Secure permissions are displayed. The Create permission is not applicable to an individual content object.

For a container (such as a folder), two sets of permissions are displayed:

- The first set of permissions affects access to the object, including the ability to add members to and remove members from the object. This set of permissions has no effect on the folder’s members.
- The second set of permissions affects the access that this object conveys to its child members. See Inheritance.

TIP  An effective access value of Not Authorized on the conveyed side of a folder’s Authorization window does not guarantee that access to child members is not authorized. A direct setting on the child or another influencing rule might provide access to the child member. For example, when you create a top-level folder, the effective access values on the conveyed side for SAS Administrators are all Not Authorized. However, SAS Administrators does have effective access to a folder that you add below the top-level folder. That access comes from a predefined rule that gives SAS Administrators access to all folders.

Provide Access

1. Open the Edit Authorization window for the target object.

2. If the principal that you want to work with is not already listed, click 🔄.  
   Note: If guest access is enabled, you must select either Add Identities or Add Guest after you click 🔄.

3. Click an effective access icon (for example, 🗝).

4. In the pop-up window, select Grant as the direct setting.
   Note: If you cannot change a direct setting, you do not have Secure permission for the current object.

5. In the Edit Authorization window, click Preview. Examine the impact of your unsaved change.
   Note: If there is a relevant prohibit setting anywhere in the system, that setting has precedence over the direct grant that you added. In that case, the effective (net) result is Not Authorized (☐), and a diamond is not displayed.

6. Click Save.

Limit Access

Any access that is not granted is implicitly denied. The preferred approach is to grant selectively and to avoid use of prohibit settings.

If you must add a prohibit setting, make sure that you do not inadvertently block your own access, particularly for the Read and Secure permissions. If you do block your own access, see Troubleshooting.

CAUTION! A prohibit setting has absolute precedence, even if a more specific grant setting exists.

Add a Condition

To provide access within a particular scope or set of circumstances, add a condition.
1 Open the Edit Authorization window for the target object.

2 If the principal that you want to work with is not already listed, click 👤.  
   **Note:** If guest access is enabled, you must select either Add Identities or Add Guest after you click 👤.

3 Click an effective access icon (for example, ☑).

4 In the pop-up window, select Conditional Grant.  
   **Note:** A conditional prohibit setting does not provide access. A conditional prohibit setting blocks all access within its scope, regardless of any more specific grant settings. A conditional prohibit setting can limit access that is provided by a grant or conditional grant setting.

5 In the Condition window, create an expression that specifies the scope and circumstances in which access is granted. Your syntax is validated when you click OK. See Rule Conditions.

6 In the Edit Authorization window, click Preview. Examine the impact of your unsaved change.

7 Click Save.

---

**Edit a Condition**

1 In the Edit Authorization window for an object, click the effective access icon for the direct conditional setting that you want to modify.

2 In the pop-up window, next to the Conditional Grant or Conditional Prohibit direct setting, click ☑️.

3 In the Condition window, edit the expression. Your syntax is validated when you click OK. See Rule Conditions.

4 In the Edit Authorization window, click Preview. Examine the impact of your unsaved change.

5 Click Save.

---

**Delete a Condition**

1 In the Edit Authorization window for an object, click the effective access icon for the direct conditional setting that you want to delete.

2 In the pop-up window, next to the Conditional Grant or Conditional Prohibit direct setting, click ☑️.

3 In the Condition window, delete the expression. Click OK.

4 In the Edit Authorization window, click Preview. Examine the impact of your unsaved change.

5 Click Save.

---

**Remove a Direct Setting**

1 Open the Edit Authorization window for an object.

2 In the cell that has the direct setting that you want to remove, click the effective access icon. In the pop-up window, select (none) as the direct setting.
   **Note:** If you cannot change the direct setting, you do not have Secure permission for the current object.
In the Edit Authorization window, click Preview. Examine the impact of your unsaved change.  
Note: Any identities that are no longer principals are automatically removed.

Click Save.

Identify the Source of Effective Access
To determine which rules and shares contribute to a particular effective access result, examine the origins information for that result.

1. Open the View Authorization window for the target content object.
2. Click the effective access icon for which you want origins information.
3. In the pop-up window, the Contributing Rules tab provides a read-only display of all applicable rules, except share-based rules. Here are tips:
   - If you opened the Edit Authorization window and have unsaved (and un-previewed) changes, the Contributing Rules tab is disabled.
   - To view additional details, add columns to the table. Click and select Manage columns.
   - To directly modify a rule, use the Rules page.
4. In the pop-up window, the Contributing Shares tab provides a read-only display of all relevant shares. Here are tips:
   - If sharing is disabled, the Contributing Shares tab is not displayed.
   - To view additional details, add columns to the table.
   - To directly modify a rule, use the Rules page.
   - To manage shares, see “Administrative Oversight on the Rules Page”.

General Authorization: How to (Rules Page)

Introduction
These instructions explain how to directly manage general authorization rules using SAS Environment Manager.

Navigation
In the applications menu ( ), under Administration, select Manage Environment. In the vertical navigation bar, select .

The Rules page is an advanced interface. It is available to only SAS Administrators. You can use a simpler interface to set permissions on content such as folders and reports.

Rules Page
Use the Rules page to manage authorization rules directly. Here are examples:

- View and filter rules.
- Enable and disable rules.
- Replace the principal in a rule.
- View and edit a rule’s description or reason.
- Use an existing rule as the basis for a new rule.
- Work with rules that affect access to functionality.
- View conditions for multiple rules at the same time.

Here are additional details:
- To ensure that all rules that should be visible to you are displayed, refresh the display and click **Reset all** in the **Rules Filter** pane.
- On the **Rules** page, you cannot see rules that are directly assigned to objects for which you lack the Secure permission.
- The search field searches only the **Object URI**, **Description**, **Reason**, and **Condition** columns.
- To add, remove, or reorder columns, click **[ ]**, and select **Manage columns**.
- You cannot sort the values within a column.
- You can use the **Rules Filter** pane to view a subset of rules. Some filters take effect immediately, other filters take effect after you click **Apply**.
- You can clear a filter by clicking its **Reset** link. You can clear all filters by clicking **Reset all** at the top of the pane.
- The **Guest** principal type is always listed, regardless of whether guest access is enabled.
- Display names for users and groups are not available on the **Rules** page.
- For details about rule attributes, see **Rule Attributes**.

### Add a Rule

1. On the **Rules** page, click **[ ]**.

2. In the New Rule window, provide values for at least the required attributes. Here are tips:
   - In some fields, you can click ![Icon](image) or ![Icon](image) to browse instead of directly entering a value.
   - If a warning indicates that the **Principal** value cannot be validated, make sure the value is an ID, not a display name. If the principal is a service account (such as sasapp or sas.folders), you can ignore the warning.
   - To populate the list of permissions, use the **Clear All**, **Select All**, and **Choose** buttons.

3. Click **Save**.

4. On the **Rules** page, right-click the new rule, and select **Properties**. Verify that the attributes of the new rule are as you intended.

5. If the rule affects a content object (such as a folder or report), use the **Authorization window** to verify that the results are as you intended.

### Edit a Rule

1. On the **Rules** page, select a rule, and then click **[ ]**.

   **Note:** You cannot edit a share-based rule. See “Sharing: Details for Administrators”.
In the Edit Rule window, modify attributes as needed. Here are tips:

- If a warning indicates that the Principal value cannot be validated, make sure the value is an ID, not a display name. If the principal is a service account (such as sasapp or sas.folders), you can ignore the warning.
- If the rule does not have a condition, an Add Condition button is present. If the rule has a condition, an Edit Condition button is displayed.

3 Click Save.

4 On the Rules page, right-click the rule, and select Properties. Verify that the attributes of the new rule are as you intended.

5 If the rule affects a content object (such as a folder or report), use the Authorization window to verify that the results are as you intended.

Copy a Rule

1 On the Rules page, select a rule, and then click Copy.
   
   Note: You cannot copy a share-based rule. See “Sharing: Details for Administrators”.

2 In the New Rule window, modify attributes as needed.

3 Click Save.

4 On the Rules page, right-click the new rule, and select Properties. Verify that the attributes of the new rule are as you intended.

5 If the rule affects a content object (such as a folder or report), use the Authorization window to verify that the results are as you intended.

Delete a Rule

1 On the Rules page, select a rule, and then click Delete.

2 In the confirmation window, click Delete.

3 If the rule affects a content object (such as a folder or report), use the Authorization window to verify that the results are as you intended.

Edit a Condition

1 On the Rules page, select a rule, and then click Edit.

2 In the Edit Rule window, click Edit Condition.
   
   Note: If a rule does not have a condition, an Add Condition button is present. If a rule has a condition, an Edit Condition button is displayed.

3 In the Edit Condition window, edit the expression. Your syntax is validated when you click OK.

4 Click Save.

5 If the rule affects a content object (such as a folder or report), use the Authorization window to verify that the results are as you intended.
Delete a Condition

1. On the Rules page, select a rule, and then click edit.

2. In the Edit Rule window, click Edit Condition.
   
   **Note:** If a rule does not have a condition, an Add Condition button is present. If a rule has a condition, an Edit Condition button is displayed.

3. In the Edit Condition window, delete the expression.

4. Click Save.

5. On the Rules page, verify that the condition no longer exists. Right-click on the rule, select Properties, and verify that the Condition field is blank.

6. If the rule affects a content object (such as a folder or report), use the Authorization window to verify that the results are as you intended.

Locate a Particular Rule

Here are general tips:

- Filter requirements are cumulative. For example, if you set two filters, only rules that meet both criteria are displayed.
- Remember to click Apply after you set or modify certain filters.
- You can search for rules that contain specified text in the ObjectUri, Description, Reason, or Condition field. The search is not case-sensitive. The search looks for the specified text in any of the four supported fields.

Here are tips for locating a rule by date:

- In the Rules Filter pane, under Date Modified, click to select a date or date range.
- Each rule’s Date Modified value indicates when the rule was created or most recently modified.
- To add the Date Modified column to the display, click , and select Manage columns.

Here are tips for locating a rule by identity:

- The Modified By filter is based on who created or last updated a rule.
- To add the Modified By column to the display, click , and select Manage columns.
- The Principal filter is based on who a rule is assigned to.
- Both the Modified By and the Principal filters use ID values, not display name values. For example, to display only rules that were created by userA, specify userA in the Modified By filter. To display only rules that are assigned to the SAS Administrators group, specify SASAdministrators in the Principal filter.
- Rules that are predefined or generated can have a Modified By value that does not correspond to a user or group that is known to the identities service.

Here are tips for locating a rule by the URI of the rule’s target:

- To view only those rules that target a specific content object, use the technique that is appropriate for the type of URI, as follows:
  - Browse content objects for the object URI for a rule target. In the drop-down list under Object URI, select URI.
Browse container objects for the container URI for a rule target. In the drop-down list under Container URI, select URI.

To view only those rules that do not specify an object URI, select (blank URI) from the drop-down list under Object URI.

To view rules that either specify /** as the object URI or do not specify an object URI, select (global URI) from the drop-down list under Object URI.

To view only those rules that do not specify a container URI, select (blank URI) from the drop-down list under Container URI.

For tips for locating share-based rules, see “Administrative Oversight on the Rules Page” on page 32.

**Extend the Ability to Create Top-Level Folders**

Initially, only SAS Administrators can create top-level folders. Here are examples of how you can use the Rules page to extend that ability to other users:

- To enable all authenticated users to create top-level folders, locate the rule that targets the object URI /folders/folders and grants the Add and Read permissions to Authenticated Users. Edit that rule so that it also grants the Create permission.

- To enable a group that has the ID groupA to create top-level folders, add a new rule that targets the object URI /folders/folders and grants the Create permission to groupA.

**General Authorization: Concepts**

**Scope**

General authorization manages access to the following resources:

- content, such as folders and reports
- functionality, such as applications, features, and services

**TIP** For information about the sharing endpoint within the authorization service, see “Sharing: Details for Administrators”.

**Key Terms**

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule</td>
<td>A composite of authorization elements. Example: A rule grants groupA the Read permission for folderA.</td>
</tr>
<tr>
<td>Target</td>
<td>The affected resource, such as an individual object, a set of objects, a service, or a service endpoint. Examples: folderA, reportA</td>
</tr>
<tr>
<td>Principal</td>
<td>The user, group, or construct to which a rule is assigned. Examples: UserA, GroupA, Authenticated Users</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Permission</td>
<td>A type of access. Values: Add, Create, Delete, Read, Remove, Secure, Update</td>
</tr>
<tr>
<td>Setting</td>
<td>In a rule, the indication of whether (and to what extent) access is provided. Values: Grant, Conditional Grant, Prohibit, Conditional Prohibit</td>
</tr>
<tr>
<td>Condition</td>
<td>In a rule, the constraint expression. Most rules do not include a condition. Example: currentUser() == #preferenceOwner</td>
</tr>
<tr>
<td>Effective access</td>
<td>A context-neutral description of the net result of all relevant rules. Effective access does not incorporate evaluation of any conditions. Values: Authorized, Not Authorized, Conditional</td>
</tr>
<tr>
<td>Access outcome</td>
<td>In a context-aware access request, the authorization decision. Values: Authorized, Not Authorized</td>
</tr>
</tbody>
</table>

**Principals**

The principal in an authorization rule is the user, group, or construct to which the rule is assigned. The general authorization system supports the following principals:

- A user is either an individual authenticated user or a service account.
- A user group is either a custom group or a group in your authentication provider.
- Authenticated Users is the principal type that represents all authenticated users.
- Everyone is the principal type that represents all principals.
- Guest is the principal type that facilitates guest access. Guest is not part of Authenticated Users, but is part of Everyone.

**Note:** When a principal is deleted, rules that are assigned to that principal are not automatically deleted. Such rules are reused if a new principal of the same type and ID is created. The general authorization system does not have an automated cleanup process for orphaned rules.

**Administrators**

The SAS Administrators group provides access throughout the general authorization system. A predefined rule grants all permissions throughout the general authorization system to the SAS Administrators group. However, the SAS Administrators group is not unrestricted or exempt from authorization requirements.

For more information, see **Predefined Custom Groups** in *SAS Viya Administration: Identity Management*.

**Inheritance**

Access flows through a hierarchy of containers. Each container conveys settings to its child members. Each child member inherits settings from its parent container. For example, a folder’s child members might include reports and other folders.

**Note:** A reference member (such as a shortcut) does not inherit access from its parent folder.
You can manage access that a container conveys independently from access to the container. Here are examples of that separation:

- In a folder’s Authorization window, the first set of permissions depicts access to the folder, and the second set of permissions depicts access that the folder conveys.
- On the Rules page, a rule that targets a folder can affect either or both types of access, depending on which fields (Object URI, Container URI, or both) are populated.

A rule that targets the object aspect of a container (the container’s objectUri attribute) has different effects than a rule that targets the container aspect of a container (the container’s containerUri attribute). Here are details, using folderA as an example container.

<table>
<thead>
<tr>
<th>Rule Target</th>
<th>Potential Impact of the Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>folderA (as an object)</td>
<td>Can affect the ability to read, update, or delete folderA.</td>
</tr>
<tr>
<td></td>
<td>Can affect the ability to add or remove members for folderA.</td>
</tr>
<tr>
<td></td>
<td>Settings are not conveyed to the objects within folderA.</td>
</tr>
<tr>
<td>folderA (as a container)</td>
<td>Settings are conveyed to folderA’s child members.</td>
</tr>
<tr>
<td>folderA (as an object and as a container)</td>
<td>Can affect the ability to read, update, or delete folderA.</td>
</tr>
<tr>
<td></td>
<td>Can affect the ability to add or remove members for folderA.</td>
</tr>
<tr>
<td></td>
<td>Container settings are conveyed to folderA’s child members.</td>
</tr>
</tbody>
</table>

Some interfaces enable you to create rules that target both aspects of access to a container. However, containerUri settings are never derived from or implicitly matched to objectUri settings. This separation enables you to provide Write access to the objects in a container without providing Write access to the container itself.

**Permissions**

<table>
<thead>
<tr>
<th>Permission</th>
<th>Affected Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create*</td>
<td>Create a new object.</td>
</tr>
<tr>
<td>Read</td>
<td>Read an object.</td>
</tr>
<tr>
<td>Update</td>
<td>Update or edit an object.</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete an object.</td>
</tr>
<tr>
<td>Secure</td>
<td>Set permissions on an object (manipulate the object’s direct rules).</td>
</tr>
<tr>
<td>Add**</td>
<td>Put an object into a container.</td>
</tr>
<tr>
<td>Remove***</td>
<td>Move an object out of a container.</td>
</tr>
</tbody>
</table>

* Applicable for a service, service endpoint, or media type.
** Applicable for a container, such as a folder.
*** Applicable for a container, such as a folder. In SAS Environment Manager, also affects the ability to delete a child member from a folder.
### Table 1  Permission Settings

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prohibit</td>
<td>Prevents access.</td>
</tr>
<tr>
<td>Conditional Prohibit</td>
<td>Prevents access in specified circumstances and scope.</td>
</tr>
<tr>
<td>Grant</td>
<td>Provides access, unless there is a relevant Prohibit or Conditional Prohibit setting.</td>
</tr>
<tr>
<td>Conditional Grant</td>
<td>Provides access in specified circumstances and scope, unless there is a relevant Prohibit or Conditional Prohibit setting.</td>
</tr>
</tbody>
</table>

**Note:** Setting is a compound of rule type and whether a condition is present. Setting is a client-layer convenience construct, not a service-layer rule attribute.

### Permissions by Task

#### Introduction

To provide sufficient access to complete a task, you must consider both the availability of functionality and the availability of individual objects. For example, here are the primary requirements for creating and saving a new report:

- the ability to create reports (for example, the Create permission for the service endpoint that controls the ability to create reports). For more information, see Access to Functionality in SAS Viya Administration: Identity Management.

- the ability to add members to the target folder (the Add permission for the object aspect of the target folder).

- the ability to see and update the new report (for example, the Read and Update permissions for the target folder’s containerUri). See Inheritance.

**Note:** In addition to the requirements that are documented in this topic, most interfaces enable you to interact with only those resources for which you have Read access.

#### Details for Top-Level Folders

Here are the required permissions for managing top-level folders:

<table>
<thead>
<tr>
<th>Task</th>
<th>Service URI</th>
<th>Top-Level Folder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a top-level folder*</td>
<td>Create</td>
<td>-</td>
</tr>
<tr>
<td>Delete a top-level folder</td>
<td>-</td>
<td>Delete</td>
</tr>
<tr>
<td>Rename a top-level folder</td>
<td>-</td>
<td>Update</td>
</tr>
<tr>
<td>Manage access to a top-level folder</td>
<td>-</td>
<td>Secure</td>
</tr>
</tbody>
</table>

* Initially, only members of the SAS Administrators group can add top-level folders. See Extend the Ability to Create Top-Level Folders.

Initial access to a new top-level folder is as follows:
The user who creates a new top-level folder has full access to that folder. Automatically generated direct grants provide that access.

SAS Administrators has full access to every new top-level folder. The predefined rule that grants SAS Administrators permissions for all objects provides that access. See Examine Access.

See also “Content Management: How To” in SAS Viya Administration: Content Management.

Details for Child Members

A child member is an object that is in a folder and is not a reference member. For example, folders (other than top-level folders) and reports are child members.

Here are the required permissions for managing child members:

<table>
<thead>
<tr>
<th>Task</th>
<th>Service URI</th>
<th>Child Member</th>
<th>Current Parent Folder</th>
<th>New Parent Folder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Create</td>
<td>(Read, Update)*</td>
<td>Add</td>
<td>-</td>
</tr>
<tr>
<td>Delete</td>
<td>-</td>
<td>Delete</td>
<td>Remove**</td>
<td>-</td>
</tr>
<tr>
<td>Update</td>
<td>-</td>
<td>Update</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Rename</td>
<td>-</td>
<td>Update</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Move</td>
<td>-</td>
<td>Update</td>
<td>Remove</td>
<td>Add</td>
</tr>
<tr>
<td>Manage access</td>
<td>-</td>
<td>Secure</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

* These permissions are required for only objects that are updated during their creation process. For example, the process of creating and saving a new report includes an internal update to the content of the new report. The necessary access is usually conveyed from the parent folder.

** This requirement applies only in SAS Environment Manager.

Initial access to a new child member is determined by inheritance and any other influencing rules, including any automatically generated direct settings.

See also “Content Management: How To” in SAS Viya Administration: Content Management.

Details for Reference Members

A reference member is a pointer to another resource. For example, an item in a list of favorite or recent objects is a reference member.

Access to a reference member is independent from access to the referenced resource. For example, having Read access to a favorite that points to ReportA does not equate to having Read access to ReportA. Conversely, having Read access to ReportA does not equate to having Read access to all reference members that point at ReportA.

Access to reference members is as follows:

- Anyone who has Read access to a folder can see all reference members in that folder.
- Anyone who has Remove access to a folder can delete all reference members in that folder.
- You cannot set permissions on a reference member.
- A reference member does not inherit permissions.

See also “Content Management: Concepts” in SAS Viya Administration: Content Management.
Details for Authorization Rules

Anyone who has the Secure permission for a resource can add, modify, and delete direct rules for that resource. In the initial configuration, the SAS Administrators group can add, modify, and delete all authorization rules.

Note: Sharing can enable users who do not have the Secure permission for an object to share that object, indirectly creating a share-based authorization rule.

Details for Models

Registered models are stored beneath the Model Repositories node in the SAS Content tree. Initially, all authenticated users have Read and Write access to all registered models. That access is provided by predefined and generated rules that grant broad access to model repositories. See “Default Permissions” in SAS Model Manager: Administrator’s Guide.


HTTP Mapping

Here are the standard mappings of HTTP verbs to permissions:

<table>
<thead>
<tr>
<th>HTTP Verb</th>
<th>Permission</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>Create</td>
</tr>
<tr>
<td>DELETE</td>
<td>Delete</td>
</tr>
<tr>
<td>GET, OPTIONS, HEAD</td>
<td>Read</td>
</tr>
<tr>
<td>PUT, PATCH</td>
<td>Update</td>
</tr>
</tbody>
</table>

Some actions override the default mappings and instead require a different permission.

Rule Targets

About Rule Targets

Each rule affects a target resource (or set of resources), as identified in the rule’s target-related attributes (objectUri, containerUri, and the media type properties). A rule can specify any, all, or none of the target-related attributes.

CAUTION! A rule that does not specify at least one target-related attribute affects access to all resources throughout the general authorization system.

About Target URIs

TIP Do not confuse uniform resource identifiers (URIs) and folder locations. Although both constructs use paths to reference resources, the two constructs are entirely independent and distinct. For example, a report might have a URI of /reports/reports/qwe3429ryjw12567 and a folder location of /SAS Content/Public/reportA.
A target URI is the value in an authorization rule’s objectUri or containerUri attribute. A target URI references a resource such as a service, service endpoint, or content object. The authorization service sees target URIs as strings. The authorization service does not know whether a particular URI represents an application, a feature, an object, or a collection of objects.

Authorization decisions for a particular resource evaluate all rules that have target URIs that match the requested URI. To compare a requested URI to the target URIs in authorization rules, the authorization service uses Ant-style pattern matching. In that pattern matching, the authorization service supports a wildcard character, so that the target URI in a rule can be more general than the target URI in an individual request. Here are details:

- A single wildcard (*) matches a single element in a URI path. For example, it matches the name of a service or the ID of an object.
- A double wildcard (**) matches any number of consecutive elements in a URI path. For example, it matches a forward slash (/), the name of a service, the ID of an object, a multi-element path (/lmn/xyz/), or the absence of any element (null).

The following tables provide examples. The first table uses fictional URIs to demonstrate the principles. The other tables use selected URIs to illustrate how the principles apply in context.

### Table 2  Scope of a Target URI: Principles

<table>
<thead>
<tr>
<th>Rule’s Target URI</th>
<th>Rule’s Relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>/**</td>
<td>Relevant to all requests.</td>
</tr>
<tr>
<td>/abc</td>
<td>Relevant to requests for exactly this URI: /abc.</td>
</tr>
<tr>
<td>/abc/</td>
<td>Relevant to requests for exactly this URI: /abc/.</td>
</tr>
<tr>
<td>/abc/*</td>
<td>Relevant to requests for URIs such as /abc/lmn (not /abc, /abc/, or URIs such as /abc/lmn/ and /abc/lmn/xyz).</td>
</tr>
<tr>
<td>/abc/*/xyz</td>
<td>Relevant to requests for URIs such as /abc/lmn/xyz (not /abc/xyz).</td>
</tr>
<tr>
<td>/abc/**</td>
<td>Relevant to requests for /abc, /abc/, and URIs such as /abc/lmn, /abc/lmn/, and /abc/lmn/xyz.</td>
</tr>
<tr>
<td>/abc/**/xyz</td>
<td>Relevant to requests for /abc/xyz, and URIs such as /abc/lmn/xyz and /abc/lmn/rst/xyz.</td>
</tr>
</tbody>
</table>

### Table 3  Scope of a Target URI: Examples from Reports Service

<table>
<thead>
<tr>
<th>Rule’s Target URI</th>
<th>Rule’s Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>/reports</td>
<td>The root of the reports service. Relevant to requests that omit a trailing slash.</td>
</tr>
<tr>
<td>/reports/</td>
<td>The root of the reports service. Relevant to requests that include a trailing slash.</td>
</tr>
<tr>
<td>/reports/*</td>
<td>All first-level endpoints within the service (not the service root, lower-level endpoints, or individual reports).</td>
</tr>
<tr>
<td>/reports/*/report-ID</td>
<td>All first-level endpoints within the service (for the specified report).</td>
</tr>
<tr>
<td>Rule’s Target URI</td>
<td>Rule’s Scope</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>/reports/**</td>
<td>All reports, all endpoints within the service, and the service root.</td>
</tr>
<tr>
<td>/reports/**/report-ID</td>
<td>All endpoints within the service (for the specified report).</td>
</tr>
<tr>
<td>/reports/reports/*</td>
<td>All reports, and all second-level endpoints beneath the reports endpoint.</td>
</tr>
<tr>
<td>/reports/reports/report-ID</td>
<td>The specified report.</td>
</tr>
</tbody>
</table>

Table 4  Scope of a Target URI: Examples from Model Repository Service

<table>
<thead>
<tr>
<th>Rule’s Target URI</th>
<th>Rule’s Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>/modelRepository</td>
<td>The root of the model repository service. Relevant to requests that omit a</td>
</tr>
<tr>
<td></td>
<td>trailing slash.</td>
</tr>
<tr>
<td>/modelRepository/</td>
<td>The root of the model repository service. Relevant to requests that include</td>
</tr>
<tr>
<td></td>
<td>a trailing slash.</td>
</tr>
<tr>
<td>/modelRepository/*</td>
<td>All first-level endpoints within the service (not the service root, lower-level</td>
</tr>
<tr>
<td></td>
<td>endpoints, or individual objects).</td>
</tr>
<tr>
<td>/modelRepository/**</td>
<td>All objects that are managed by the service (repositories, projects, and</td>
</tr>
<tr>
<td></td>
<td>models), all endpoints within the service, and the service root.</td>
</tr>
<tr>
<td>/modelRepository/repositories/*</td>
<td>All repositories, and all second-level endpoints that are beneath the</td>
</tr>
<tr>
<td></td>
<td>repositories endpoint.</td>
</tr>
<tr>
<td>/modelRepository/projects/*</td>
<td>All projects, and all second-level endpoints that are beneath the projects</td>
</tr>
<tr>
<td></td>
<td>endpoint.</td>
</tr>
<tr>
<td>/modelRepository/models/*</td>
<td>All models, and all second-level endpoints that are beneath the models</td>
</tr>
<tr>
<td></td>
<td>endpoint.</td>
</tr>
<tr>
<td>/modelRepository/models/model-ID</td>
<td>The specified model.</td>
</tr>
</tbody>
</table>

**TIP**  As the preceding examples make clear, not all services have identical structures. Before you create or customize a rule that uses a wildcard in its target URI, make sure you understand the structure of the service, and the resulting scope of the rule. (For services that have published APIs, you can use developer.sas.com to discover structure.)

**objectUri**

A rule that targets an objectUri affects access to the referenced resource. Here are examples:

- A rule that targets a folder’s objectUri affects access to that folder.
- A rule that targets a report’s objectUri affects access to that report.
- A rule that targets a service’s objectUri (and does not target a specific object instance) affects access to functionality. See Access to Functionality in SAS Viya Administration: Identity Management.
In general, rules that specify the objectUri for a content object (such as a folder or report) should include the /** suffix. Inadvertently omitting the suffix narrows the effects of a rule and can yield unintended results due to insufficient access. Here are details:

- The Authorization window appends the /** suffix to new rules that target objectUris. You cannot see or modify URIs in the Authorization window.

- The New Rule window appends the /** suffix when you populate the Object URI field by clicking and selecting a content object in the Choose an Item window. You can see and modify URIs in the New Rule and Edit Rule windows.

- In other interfaces and contexts, you must remember to include a suffix, when appropriate.

**containerUri**

A rule that targets a folder’s container URI affects access that the folder conveys to its child members.

**Note:** You cannot append the /** suffix to a containerUri. The /** suffix does not reference contained objects (such as reports within a folder). The /** suffix has nothing to do with container-based object inheritance.

**mediaType**

This property has no effect and is deprecated. It is replaced by the more specific properties acceptType, acceptItemType, and contentType.

**acceptType, acceptItemType, and contentType**

The three media type properties (acceptType, acceptItemType, and contentType) provide specialized, advanced support for refining the scope of a rule. Most rules do not specify these attributes. To view or set these attributes, use the REST API.

You can use media type properties to limit a rule’s applicability so that it targets only a particular media type in only a specified context.

- If a rule specifies one media type property, the rule applies only to requests that specify a corresponding value for the corresponding attribute.

  For example, if a rule specifies contentType = 'application/pdf' then the rule applies only to requests in which the following query parameter is specified: (contentType == application/pdf).

- If a rule specifies multiple media type properties, the rule applies only to requests that specify corresponding values for all specified media type properties.

  For example, if a rule specifies the value application/pdf in both the contentType and the acceptType properties, then the rule applies to only requests for which the following compound set of parameters is specified: (contentType == application/pdf and acceptType == application/pdf).

These properties are processed as follows:

- In response to an authorization request that specifies one or more of the media type query parameters (contentType, acceptType, or acceptItemType), only matching rules are applied. A troubleshooting item provides an example.

- In response to an authorization request that does not specify a media type query parameter (contentType, acceptType, or acceptItemType), any rules that specify one or more media type properties are ignored.
# Rule Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Target-related attributes:</strong></td>
<td></td>
</tr>
<tr>
<td>objectUri</td>
<td>A relative URI that represents a resource such as a report, a folder, a service, or a service endpoint. Character limit: 500</td>
</tr>
<tr>
<td>containerUri</td>
<td>A relative URI that represents the container aspect of a container, such as a folder. Rules that specify a containerUri affect access that a container conveys to its child members. Character limit: 500</td>
</tr>
<tr>
<td>mediaType</td>
<td>This property is deprecated.</td>
</tr>
<tr>
<td>acceptType, acceptItemType, contentType</td>
<td>See “acceptType, acceptItemType, and contentType”.</td>
</tr>
<tr>
<td><strong>Principal-related attributes:</strong></td>
<td></td>
</tr>
<tr>
<td>principalType</td>
<td>Three of the values (Authenticated Users, Everyone, and Guest) represent classes of users. Everyone includes all authenticated users and any guest users.</td>
</tr>
<tr>
<td></td>
<td>- Assign broad grant rules to Authenticated Users.</td>
</tr>
<tr>
<td></td>
<td>- Do not assign prohibit rules to Everyone or to Authenticated Users. Such rules block access for all users, including yourself.</td>
</tr>
<tr>
<td>principal</td>
<td>The unique string that identifies a particular user or group by its ID. If principalType is user or group, you must specify a value for this attribute. Character limit: 100</td>
</tr>
<tr>
<td><strong>Access-related attributes:</strong></td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>The indication of whether a rule blocks (prohibit) or attempts to provide (grant) access. Prohibit rules have absolute precedence.</td>
</tr>
<tr>
<td>condition</td>
<td>An expression that limits the scope or applicability of a rule. Character limit: 5120</td>
</tr>
<tr>
<td>permissions</td>
<td>A list of access types. At least one permission is required.</td>
</tr>
<tr>
<td>enabled</td>
<td>The indication of whether a rule is enabled. By default, rules are enabled. To temporarily prevent a rule from being enforced, disable the rule.</td>
</tr>
<tr>
<td><strong>Documentation-related attributes:</strong></td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>Text that documents a rule for administrative purposes. Character limit: 1000</td>
</tr>
<tr>
<td>Attribute</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>reason</td>
<td>Text that provides information for end users, where supported by a client. For example, a prohibit rule’s reason could be displayed to an end user as part of an access denied message. Character limit: 1000</td>
</tr>
</tbody>
</table>

* In SAS Environment Manager, this attribute is labeled Rule Status, and has values of Enabled and Disabled.

**Rule Conditions**

**Overview**

A condition is a Boolean expression that limits the scope of a rule.

- A rule that has no condition is always applied.
- A rule that has a condition that evaluates to true for a particular access request is applied in the authorization decision process for that access request.
- A rule that has a condition that evaluates to false for a particular access request is ignored in the authorization decision process for that access request.
- If a rule has an invalid condition, an error is logged and access is restricted as follows:
  - If a grant rule has an invalid condition, the rule is always ignored.
  - If a prohibit rule has an invalid condition, the rule is always applied.

You can specify a condition in inclusive or exclusive terms. Here are two examples:

- A rule grants the Read permission to GroupA for folderA, with a condition that the rule applies only on weekdays. A request from a member of GroupA to access folderA on Sunday is outside the condition. For that access request, the condition is false, so the rule is not applicable (it does not provide access).
  
  **Note:** A conditional grant rule provides access in specified circumstances, but it does not prevent access outside of those circumstances.

- A rule prohibits the Read permission for GroupA for folderA, with a condition that the rule does not apply on weekends. A request from a member of GroupA to access the folder on Sunday is inside the condition. For that access request, the condition is true, so the rule is applicable (access is blocked).
  
  **Note:** A conditional prohibit rule prevents access in specified circumstances, but it does not provide access outside of those circumstances.

**Condition Syntax**

- Conditions are written and stored in Spring Expression Language (SpEL).
- Boolean operators (AND, OR, and NOT) and parentheses are supported. For example, the following condition always evaluates to true:
  
  $\{4 < 6\} \text{ and } \{5 > 3\}$

- Built-in functions correspond to attributes of the requestor or the environment. You must append () to each built-in function (for example, currentUser()).
  
  **Note:** You can use a constant instead of a function. However, functions are often more useful because they are dynamic. At request time, actual context-specific values are dynamically substituted into each function.
Variables correspond to attributes of the target. You must prepend # to each variable (for example, 
userId).

Note: The available condition variables for a particular type of object are designated in the service for that 
object type. For example, the preference service designates the userId attribute on preference objects as 
available for use as a condition variable.

Built-In Functions

Table 5  Location-Based Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>locale()</td>
<td>Locale of the client that made the request (for example, en_US).</td>
<td>String</td>
</tr>
<tr>
<td>remoteHost()</td>
<td>Name of the client machine that made the request.</td>
<td>String</td>
</tr>
<tr>
<td>remoteIp()</td>
<td>IP address of the client machine that made the request.</td>
<td>String</td>
</tr>
<tr>
<td>serverIp()</td>
<td>IP address of the middle-tier server that received the request.</td>
<td>String</td>
</tr>
<tr>
<td>serverName()</td>
<td>Machine name of the middle-tier server that received the request.</td>
<td>String</td>
</tr>
<tr>
<td>serverPort()</td>
<td>Port of the middle-tier server that received the request.</td>
<td>int</td>
</tr>
</tbody>
</table>

Table 6  Target-Based Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>contentType()</td>
<td>Content type of the target (for example, application/vnd.sas.credential.domain+json).</td>
<td>String</td>
</tr>
<tr>
<td>contentLength()</td>
<td>Length of the request.</td>
<td>long</td>
</tr>
<tr>
<td>uri()</td>
<td>URI of the target.</td>
<td>String</td>
</tr>
</tbody>
</table>

Table 7  Time-Based Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>timestamp</td>
<td>Coordinated Universal Time (UTC) timestamp.</td>
<td>ZonedDateTime</td>
</tr>
<tr>
<td>timestamp(zoneId)</td>
<td>Timestamp of the request, based on a specified zoneid.*</td>
<td>ZonedDateTime</td>
</tr>
<tr>
<td>localTime(zoneId)</td>
<td>Time of the request, based on a specified zoneid.*</td>
<td>LocalDateTime</td>
</tr>
<tr>
<td>localDate(zoneId)</td>
<td>Date of the request, based on a specified zoneid.*</td>
<td>LocalDate</td>
</tr>
<tr>
<td>localDateTime(zoneId)</td>
<td>Date and time of the request, based on a specified zoneid.*</td>
<td>LocalDateTime</td>
</tr>
</tbody>
</table>

* A time zone ID that is valid for java.time.ZonedDateTime.
### Table 8  Other Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>currentUser()</td>
<td>Identifier for the currently connected user.</td>
<td>String</td>
</tr>
<tr>
<td>groupsForCurrentUser()</td>
<td>Identifier for each group to which the current user belongs. Nested memberships are included. Unassumed memberships are not included.</td>
<td>List</td>
</tr>
<tr>
<td>method()</td>
<td>Method that the request invoked (for example, GET).</td>
<td>String</td>
</tr>
<tr>
<td>protocol()</td>
<td>Protocol of the request (for example, HTTP/1.1).</td>
<td>String</td>
</tr>
<tr>
<td>header(headerName)</td>
<td>Headers for a specified headerName.</td>
<td>List</td>
</tr>
</tbody>
</table>

### Examples of Conditions

This condition makes its associated rule applicable only for weekday requests (in the US Eastern time zone):

```
localDate('US/Eastern').dayOfWeek != T(java.time.DayOfWeek).SUNDAY and
localDate('US/Eastern').dayOfWeek != T(java.time.DayOfWeek).SATURDAY
```

This condition makes its associated rule applicable only if the target's user ID is the same as the requesting user's ID:

```
#userId == currentUser()
```

### Evaluation of Conditions

Here are key points about evaluation of conditions:

- If a request does not meet the criteria in a condition, the request is outside that condition. If a request meets the criteria in a condition, the request is inside that condition.
- In a description of effective access, there is no request context, so conditions are not evaluated. Even an atypical condition that is always true (1=1) or never true (1>2) yields an effective access result of Conditional in certain scenarios. A condition is evaluated only in the context of a specific request.
- In an actual request, there is a request context. Any relevant conditions are evaluated, and a definitive answer is provided (Authorized or Not Authorized).

### Authorization Decisions

#### Precedence

In the general authorization system, precedence is extremely flat. The only factor that affects precedence is the type of rule (grant or prohibit). Prohibit rules have absolute precedence. If there is a relevant prohibit rule, effective access is always Not Authorized.

Neither object inheritance nor identity hierarchy has precedence implications. Here are examples:

- A grant setting that is assigned to you has less precedence than a prohibit setting that is assigned to Authenticated Users.
- A direct grant on a report has less precedence than a prohibit setting that the report inherits from its parent folder.
Cheat Sheet

In the following table, each row indicates the effective access answer for a separate, independent scenario. For example, if the only relevant rule is a Conditional Prohibit, the effective access answer is Not Authorized (because there is no relevant grant setting).

<table>
<thead>
<tr>
<th>All Relevant Rules</th>
<th>Effective Access and Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(none)</td>
<td>Not Authorized (implicit). Any access that is not granted is implicitly denied.*</td>
</tr>
<tr>
<td>Prohibit</td>
<td>Not Authorized. A relevant prohibit setting blocks access.**</td>
</tr>
<tr>
<td>Prohibit + (any other rules)</td>
<td>Not Authorized. A relevant prohibit setting has absolute precedence.</td>
</tr>
<tr>
<td>Conditional Prohibits</td>
<td>Not Authorized. No relevant grants, no access.</td>
</tr>
<tr>
<td>Grant</td>
<td>Authorized. A relevant grant provides access, if there are no relevant prohibit settings.</td>
</tr>
<tr>
<td>Grant + Conditional Grants</td>
<td>Authorized. Relevant grants provide cumulative access, if there are no relevant prohibit settings.</td>
</tr>
<tr>
<td>Grant + Conditional Prohibits</td>
<td>Conditional. Authorized for requests that are outside all of the prohibit conditions. Prohibit wins, but only within its scope.</td>
</tr>
<tr>
<td>Conditional Grants</td>
<td>Conditional. Authorized for requests that are inside any of the grant conditions.</td>
</tr>
<tr>
<td>Conditional Grants + Conditional Prohibits</td>
<td>Conditional. Authorized for requests that are outside all of the prohibit conditions and inside at least one grant condition.</td>
</tr>
</tbody>
</table>

*  This result is due to the lack of a grant, so you can override it by adding a grant.
** This result is due to a prohibit setting, so you cannot override it by adding grants. As long as the prohibit setting exists and is relevant, effective access is not authorized.

For details about conditional access, see Evaluation of Conditions.

Origins of Effective Access

An explanation of effective access answers the question, Why does this identity have this effective access result for this permission and object?

In general authorization, the explanation consists of these items:

- A list of contributing rules, which includes all relevant rules, except share-based rules. Rules that are relevant but not determinative are included.
A list of contributing shares, which includes all relevant shares.
To view origins information in the Authorization window, see Identify the Source of Effective Access.

---

**General Authorization: Guidelines**

The following basic guidelines contribute to simplicity and security.
- Minimize use of prohibit rules.
- Limit membership in administrative groups.
- Use groups, not individual users, as principals.
- Use folders, not individual objects, as targets.
- Use conditions only if you cannot efficiently express your authorization requirements another way.
- Perform a backup before and after you make significant changes to your system.

---

**General Authorization: Troubleshooting**

**Unexpected Outcomes**

Here are tips for troubleshooting an authorization outcome that differs from what you expect:
- Make sure all relevant rules are enabled. On the Rules page, right-click a rule to view all of its properties. Or, add the Rule Status column to the display.
- Make sure you understand the precedence model. See Authorization Decisions.
- Examine the origins information for the unexpected outcome. See Identify the Source of Effective Access.
- If the unexpected outcome relates to inheritance from a folder to objects in that folder, make sure you are using the second set of permissions in the folder’s Authorization window to convey access to the folder’s child members. See Inheritance.
- If the unexpected outcome relates to your access, and you have changed your memberships in the current session, sign out and then sign back in.
- If the unexpected outcome relates to your access, and you are a member of the SAS Administrators group, sign out and then sign back in, indicating whether you want that membership to be in effect. See Impact of Assumable Memberships.
- If the unexpected outcome is for access to a caslib or table, see Cloud Analytic Services Authorization.

---

**Unavailable Principals**

To grant access to a principal that is not in the identities service, use the Rules page in SAS Environment Manager or use the command-line interface.

---

**Unrecognized Principals**

If the Rules page or the Authorization window displays a warning icon next to a principal’s name, that principal does not exist in the identities service.
If the principal is a service account (for example, sas.folders or sasapp), you can ignore the warning icon.

If the principal is a non-existent group that has administrators as its name and ID, the rule has no effect. Determine whether the rule should be modified or deleted.

For example, a predefined wildcard rule targets the objectURI /modelRepository/** and grants full access to administrators. The rule is unnecessary, because the SAS Administrators group has a universal grant. You can delete the rule or leave it in place.

If you are using the New Rule or Edit Rule window, make sure that the correct value is selected in the principal type field and the principal's unique identifier (not display name) is specified.

If you are using the Authorization window, make sure the identity still exists.

Note: Deletion of a custom group does not cause automatic deletion of rules in which that custom group is the principal.

Unintended Loss of Access

Reinstate Access: Instructions for Users
If you inadvertently block your own access to a resource, contact an administrator for assistance.
Note: Anyone who still has secure access to the blocked resource can reinstate your access.

Reinstate Access: Instructions for Administrators
To reinstate access that is blocked by a prohibit rule, complete the following steps:

1. Opt in to your assumable membership in the SAS Administrators group.

2. Try to reinstate access by disabling, modifying, or deleting the prohibit rule. Here are some tips:
   a. If the resource is a content object (such as a folder or report) and you cannot see the resource on the Content page, you lack Read access to the resource. Use the Rules page.
   b. If the resource is a content object and you cannot make changes in the resource's Authorization window, you lack secure access to the resource. Either delete the resource (if you have Delete access and the resource is not already in use) or proceed to the next step.
   c. If the resource is not a content object, use the Rules page.
   d. If you know who created (or last modified) a problematic rule or when a problematic rule was created (or last modified), use the Modified By or Date Modified filter on the Rules page to locate the problematic rule.

If you cannot reinstate access, proceed to the next step.

3. To temporarily prevent users other than yourself from using the deployment, close current sessions for users other than yourself, and disallow new sessions. See Disable Logins in SAS Viya Administration: Authentication.

4. Temporarily disable self-enforcement of authorization requirements for the authorization service.
   a. In the configuration definition for the authorization service, add a supplemental property named remote with a value of false.
     Note: To learn how to set configuration properties, see How To Configure Services in SAS Viya Administration: Configuration Properties.
   b. Restart the authorization service.
Note: To learn how to restart services, see Operate in SAS Viya Administration: General Servers and Services.

5 Disable, modify, or delete the problematic rule or rules.

6 Enable self-enforcement of authorization requirements for the authorization service.
   a In the configuration definition for the authorization service, remove the supplemental property named remote.
   b Restart the authorization service.

7 Verify that access is reinstated.

8 Make the deployment available again by allowing new user sessions.

If you cannot reinstate access, contact SAS Technical Support for assistance.

A Deleted Rule Reappears

Some of the predefined rules are bootstrapped by their associated service. If you delete one of those rules, it reappears the next time the service starts. Modifications that you make to such rules are preserved. If you are sure you do not want one of those rules to be in effect, disable that rule (instead of deleting it).

Inability to Browse to an Object

If you have access to a content object but cannot browse to that object, you lack Read access to at least one of the object’s parent folders.

To access the object, use an alternate navigation method. Here are examples:

- Access the object from a direct link or reference.
- Search for the object, and open the object from the search results list.
- If the object was shared with you, access the object from the Shared with me pane in SAS Drive. (On the All tab, select Shared. In the center pane, make sure Shared with me is selected.)

Not All Contributing Rules are Visible

If the number that is displayed on the Contributing Rules tab exceeds the number of visible listed rules, you are not currently authorized to see all contributing rules.

For the most comprehensive view, assume membership in the SAS Administrators group when you sign in.

See “Identify the Source of Effective Access”.

A Pair of Predefined Rules is Difficult to Interpret

The search service makes use of two predefined rules that might appear to block Read access to all resources for most authenticated users. No administrative action or adjustment is needed, because the rules do not actually have that effect.

Each of the two rules includes a specialized property that is not displayed in SAS Environment Manager. Those specialized properties, acceptType and acceptItemType, limit the applicability of the rules. The rules affect only requests that involve usage of the search service’s index. The rules ensure that users are not offered search results that include resources that they are unable to access.

Here are the key properties of one of the rules:
<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>objectUri</td>
<td>/*</td>
</tr>
<tr>
<td>principalType</td>
<td>authenticatedUsers</td>
</tr>
<tr>
<td>type</td>
<td>prohibit</td>
</tr>
<tr>
<td>permissions</td>
<td>read, create</td>
</tr>
<tr>
<td>condition</td>
<td>!(groupsForCurrentUser().contains('SASAdministrators')</td>
</tr>
<tr>
<td>description</td>
<td>Prevent ordinary users from getting a collection of indexable data when requesting a collection of vnd.sas.search.indexable.data.</td>
</tr>
<tr>
<td>acceptItemType*</td>
<td>application/vnd.sas.search.indexable.data+json</td>
</tr>
</tbody>
</table>

* This advanced property is not displayed in SAS Environment Manager. It limits the applicability of the predefined rule. See "acceptType, acceptItemType, and contentType".

## General Authorization: Interfaces

All general authorization requirements and constraints are always fully enforced. However, not all interfaces expose all general authorization features.

In the following table, the shaded part of each circle is an approximation of the amount of general authorization functionality that a particular interface exposes. The shading indicates relative coverage. The shading does not indicate alignment of coverage across interfaces.

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST API</td>
<td>The REST interface for general authorization.</td>
</tr>
<tr>
<td>Rules page</td>
<td>The advanced enterprise graphical interface for managing rules directly.</td>
</tr>
<tr>
<td>Authorization window</td>
<td>The basic enterprise graphical interface for managing access to content such as folders and reports.</td>
</tr>
<tr>
<td>Command-line interface</td>
<td>A simple, scriptable interface for managing access to objects and resources.</td>
</tr>
<tr>
<td>Share Window</td>
<td>A SAS Drive interface for simple sharing of content objects.</td>
</tr>
</tbody>
</table>
Sharing: Details for Administrators

Introduction

The purpose of sharing is to help users make content available to one another. This topic documents administrative aspects of the implementation of sharing that is provided by the authorization service. For usage information, see “Share” in SAS Drive: Getting Started.

The authorization service’s implementation of sharing is separate and distinct from all of the following features:

- sharing of generic (formerly SAS Data Management) projects, which is achieved by assigning owners and members. See “Projects” in SAS Drive: Getting Started.
- sharing of reports in SAS Visual Analytics, which does not affect access. See “Sharing Reports and Objects with Other Users” in SAS Visual Analytics: Designing Reports.

Who Can Share?

In the initial configuration, sharing is available as follows:

- Any user who has Secure access to an object can share that object with other users and groups.
- Any user with whom an object has been shared can further share that object, passing along some or all of the access that they received. Only the initiator of a chain of sharing has to have the Secure permission on the shared object.

For example, after UserA shares reportA with UserB, UserB can share reportA with other users, even if there is no authorization rule that grants UserB the Secure permission for reportA. (However, if an authorization rule explicitly prohibits the Secure permission for UserB on reportA, UserB cannot further share reportA.)

For a more restrictive configuration, see “How to Prevent All Sharing” or “How to Limit Re-sharing”.

What is a Share?

A share is of a set of attributes that is backed by a corresponding authorization rule.

When users share objects in SAS Drive, shares and corresponding share-based rules are generated. Each share specifies a particular type of access, a recipient user or group, and a target object. Each corresponding share-based rule translates the share information into an authorization rule. In authorization decisions, share-based rules are evaluated in the same way as other authorization rules.

Here are details about shares:

- Shares can only expand access. Users can remove shares, but they cannot use shares to prevent access.
- Shares do not provide guaranteed or comprehensive access.
  - A share does not provide access if there is a relevant Prohibit rule.
  - A share of one object does not provide access to related resources (such as data or linked reports), parent objects (such as folders), or embedded objects (such as images or files).
Because sharing an object does not provide access to the object’s parent folders, a share recipient might not be able to navigate to the shared object. For alternatives, see “Inability to Browse to an Object” on page 26.

- Shares cannot provide variations of access that differ from the defined share types.
- Shares cannot be conditional.
- Shares cannot target a media type.
- Shares for a container object always target both the object’s URI and the object’s container URI.
- Shares are static. For example, a share of reportA by UserA to UserB is unaffected by any subsequent loss of UserA’s access to reportA.

### Attributes of a Share

- **resourceUri** specifies the URI of the content object that is shared. Also referred to as the share target.
- **sharedBy** specifies the ID of the user that created the share.
- **sharedWith** specifies the ID of the user or group that receives the share. Also referred to as the share recipient.
- **sharedWithTyp** specifies the type of identity that receives the share.
- **type** specifies the type of access that the share provides. Also referred to as the share level.
  - Users select either **Can read** or **Can read and edit** when they create a share.
  - The service stores one of the following values for each share: read, readEdit, readShare, readEditShare.
    - Note: The appended Share indicates that a share allows further sharing.

### Effects by Share Type

The effects of a share are determined by whether the shared object is a container and what type of access the share provides. The following tables provide details:

#### Table 10  Effects of Sharing a Non-Container Object

<table>
<thead>
<tr>
<th>Share Type</th>
<th>Effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>read</td>
<td>Corresponds to a grant of the Read permission.</td>
</tr>
<tr>
<td>readShare</td>
<td>Corresponds to a grant of the Read permission. Also provides the ability to further share Read access.</td>
</tr>
<tr>
<td>readEdit</td>
<td>Corresponds to grants of the Read, Update, and Delete permissions.</td>
</tr>
<tr>
<td>readEditShare</td>
<td>Corresponds to grants of the Read, Update, and Delete permissions. Also provides the ability to further share some or all of the same access.</td>
</tr>
</tbody>
</table>
Table 11  Effects of Sharing a Container Object

<table>
<thead>
<tr>
<th>Share Type</th>
<th>Effects*</th>
</tr>
</thead>
<tbody>
<tr>
<td>read</td>
<td>Corresponds to a grant of the Read permission on the container’s object URI and container URI.</td>
</tr>
<tr>
<td>readShare</td>
<td>Corresponds to a grant of the Read permission on the container’s object URI and container URI. Also provides the ability to further share the same access.</td>
</tr>
<tr>
<td>readEdit</td>
<td>Corresponds to grants of all permissions except Secure on the container’s object URI and container URI.</td>
</tr>
<tr>
<td>readEditShare</td>
<td>Corresponds to grants of all permissions except Secure on the container’s object URI and container URI. Also provides the ability to further share some or all of the same access.</td>
</tr>
</tbody>
</table>

* Sharing a container object automatically provides conveyed access to child members.

Note: The ability to re-share is not inherited. For example, the recipient of a readShare on a folder can re-share the folder, but cannot re-share individual items within that folder.

How to Limit Re-sharing

Implications
Initially, downstream sharing is supported, even for users who lack Secure access to the object that is being shared. Disabling re-sharing has the following effects:

- New shares have values of read or readEdit, not readShare or readEditShare.
- Existing shares retain their values of readShare or readEditShare. However, the Share part of those values has no effect. Only those users who have Secure access to a target object can share that object.
- Any access that was already extended through re-sharing remains in effect. Disabling re-sharing does not revoke access that already exists.
- In the Authorization window, no sharing-related information is displayed for the Secure permission.
- There is no Contributing Shares tab for the Secure permission.

Instructions
To limit downstream sharing to only those users who have Secure access to the target object:

1. In the vertical navigation bar in SAS Environment Manager, select 🏛.
2. From the View drop-down list, select Definitions.
3. In the list of definitions, select sas.authorization.
4. If no configurations exist, click New Configuration. Otherwise, select an existing configuration to edit.
5. In the New sas.authorization Configuration window (or the Edit sas.authorization Configuration window), set the reshareEnabled property to off.
Click **Save**. Your change takes effect within 30 seconds. You do not have to restart the authorization service.

7 Verify the result as follows:
   a Add a folder beneath your My Folder.
   b Share that folder with another user.
   c Ask that user to sign in (without assuming any administrative privileges) and make sure they cannot share the folder that you shared with them.

### How to Prevent All Sharing

**Implications**

Initially, sharing is enabled. Disabling sharing has the following effects:
- No new shares can be created.
- All existing share-based rules are disabled.
- In the Authorization window, no sharing-related information is displayed.
- SAS Drive hides all actions and information that are related to sharing.
- SAS Drive provides direct access to the Authorization window for authorized users.

**Instructions**

To disable sharing:

1 In the vertical navigation bar in SAS Environment Manager, select `sas.authorization`.
2 From the **View** drop-down list, select **Definitions**.
3 In the list of definitions, select **sas.authorization**.
4 If no configurations exist, click **New Configuration**. Otherwise, select an existing configuration to edit.
5 In the New sas.authorization Configuration window (or the Edit sas.authorization Configuration window), set the sharingEnabled property to **Off**.
6 Click **Save**. Your change takes effect within 30 seconds. You do not have to restart the authorization service.

7 Verify the result as follows:
   - In SAS Drive, make sure the **Share** action is not available.
   - In SAS Drive, make sure that displays such as **Shared with me** and **Items I've shared** displays are suppressed.
   - In SAS Environment Manager, examine the properties of a share-based rule. Make sure the rule is disabled.
Administrative Oversight on the Rules Page

As an administrator, you can view and delete share-based rules on the Rules page in SAS Environment Manager. When you delete a share-based rule, the corresponding share is automatically deleted along with the rule.

On the Rules page, the following constraints apply to managing share-based rules:

- You cannot add share-based rules. Sharing is primarily a user-driven activity. Share-based rules are automatically created when users share content in SAS Drive.
- You cannot edit or copy share-based rules. Any changes to share-based rules must be coordinated with changes to the associated shares, and must conform to requirements that are specific to share-based rules.

All share-based rules are created with generated text in the Description field. Here is an example:

The user "userA" shared an object with the specified principal. (This is a share-based rule.)

On the Rules page, you can filter for share-based rules as follows:

- To display only share-based rules, enter the following text in the Description filter:

  This is a share-based rule

- To display only share-based rules for a particular share recipient, specify the preceding Description filter and specify the share recipient in the Principal filter.

- To display only share-based rules that were generated by a particular user, include that user’s ID in the Description filter text as follows:

  The user "userID" shared an object with the specified principal

**TIP** You cannot instead use the Modified by filter, because all share-based rules are generated with a Modified by value of sas.authorization. Do not assume that all rules that have a Modified by value of sas.authorization are share-based rules.

Integrated View in the Authorization Window

Effective Access

In the Authorization window, effective access information reflects shares as follows:

- Effective access information reflects any access that is provided by shares.
- In the Secure and Secure (convey) columns, the Share icon < indicates that sharing is possible even though Secure access is not granted.

  **Note:** If Secure access is granted, the Share icon is not displayed. The ability to share is inherent in Secure access. See “Examine Access” on page 3.

Direct Settings

In the Authorization window, a share is not considered a direct setting.

A share does not cause a diamond to be displayed.

You cannot modify shares in the Authorization window.
Contributing Shares

After you click an effective access icon in the Authorization window, a pop-up window that includes a Contributing Shares tab is displayed. The tab provides a read-only list of the shares that are relevant to the selected effective access result.

A share is relevant if it meets all of the following criteria:

- The share specifies the current object (or a parent of that object) as the target.
- The share specifies the current principal (or a group to which that principal belongs) as the recipient.
- The share type is relevant for the selected permission. For example, for the Update and Delete permissions, the readEditShare and readEdit types are relevant, but the read and readShare types are not relevant.

Note: For the Secure permission, the readShare and readEditShare types are treated as relevant only because those types provide the ability to reshare. Sharing never creates a grant of the Secure permission.

Note: To add the Share Type column to the display, click and select Manage columns.