Principals .............................................................. 13
Administrators ......................................................... 14
Inheritance ............................................................. 14
Permissions ............................................................ 14
Permissions by Task .................................................. 15
Rule Targets ............................................................. 17
Rule Attributes ........................................................ 18
Rule Conditions ....................................................... 19
Authorization Decisions ............................................. 22

General Authorization: Guidelines ................................. 24

General Authorization: Troubleshooting ........................ 25
  Unexpected Outcomes ............................................. 25
  Unrecognized Principals ........................................... 25
  Unintended Loss of Access ....................................... 25

General Authorization: Interfaces ................................. 27
  Interfaces to General Authorization ............................ 27
General Authorization: Overview

Note: A programming-only deployment does not use the general authorization system.

To learn about the general authorization system, see General Authorization: Concepts.

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

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

Both interfaces manage authorization rules using SAS Environment Manager. The Rules page is an advanced interface for only SAS Administrators.
General Authorization: How To (Authorization Window)

Introduction
These instructions explain how to set permissions on content (such as folders, reports, and plans) using SAS Environment Manager.

Navigation
1 In the side menu (≡), under SAS Environment Manager, click Content.

TIP Do not begin by clicking Users. Authorization information is not displayed in a user or group definition.

2 Locate and select the object.
3 Right-click, and select Authorization.

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

- Authorized
- Not Authorized
- Conditional
- Unknown

Note: A diamond 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.

Here are additional details:

- There is always a row for Authenticated Users. There is also 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.
- You cannot remove a principal from the display. Only principals that are assigned to at least one rule that affects access to the current object persist in 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 target object’s Authorization window.
2. If the principal that you want to work with is not already listed, click †.
   
   **Note:** If you cannot add an identity, you do not have Secure permission for the current object.

   **TIP** In most cases, principals are groups. Users get their access through group memberships. Rules for the principal Authenticated Users are applicable to all authenticated users.

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 cell that you modified, notice that the effective access is unknown (☐). Click ☑️ to save your changes and refresh the display. Notice that effective access is known.
   
   **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. If you modified access for a group, review the impact on other principals. For example, a direct grant for GroupA affects all members of GroupA who do not have a relevant prohibit setting.

7. When you are finished making changes, close the Authorization window.

### 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 General Authorization: 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 target object’s Authorization window.
2. If the principal that you want to work with is not already listed, click +.

**TIP** In most cases, principals are groups. Users get their access through group memberships. Settings for the principal **Authenticated Users** are applicable to all users who prove their identity.

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 Authorization window, notice that effective access for the cell that you modified is unknown (☐). Click ☐ to save your changes and refresh the display. Notice that effective access is now known.

7. If you modified access for a group, review the impact on other principals.
8. When you are finished making changes, close the Authorization window.

Edit a Condition

1. In the 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 Authorization window, notice that effective access for the cell that you modified is unknown (☐). Click ☐ to save your changes and refresh the display. Notice that effective access is now known.
5. If you modified access for a group, review the impact on other principals.
6. When you are finished making changes, close the Authorization window.

Delete a Condition

1. In the 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 Authorization window, notice that effective access for the cell that you modified is unknown (○). Click ☑ to save your changes and refresh the display. Notice that effective access is known.

5 If you modified access for a group, review the impact on other principals.

6 When you are finished making changes, close the Authorization window.

Remove a Direct Setting

1 Open the 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.

3 In the Authorization window, notice that effective access for the cell that you modified is unknown (○). Click ☑ to save your changes and refresh the display. Notice that effective access is known.
   
   Note: When you save or refresh the display, any identities that are not principals in a relevant setting are removed.

4 If you modified access for a group, review the impact on other principals.

5 When you are finished making changes, close the Authorization window.
General Authorization: How To (Rules Page)

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

Navigation
In the side menu (Ξ), under SAS Environment Manager, select Security ⇒ Rules.
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, reports, and plans.

Rules Page
Use the Rules page in SAS Environment Manager 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:
- On the Rules page, you cannot see rules that are directly assigned to objects for which you lack the Secure permission.
- To add, remove, or reorder columns, click ☰, and select Columns.
- To sort a column, right-click its header.
- To view a subset of rules, select a field in the Filter by column and supply filter information. See Tips for Locating a Rule on page 10.
- To clear an Object URI or Container URI filter, delete the filter text.
- To ensure that all rules that should be visible to you are listed, refresh the display and make sure no filter is in effect.
- 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 ☰ 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**, 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, report, or plan), 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 **Edit**.

2. 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, report, or plan), 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**.

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, report, or plan), 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, report, or plan), 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, report, or plan), 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, report, or plan), use the **Authorization window** to verify that the results are as you intended.

**Tips for Locating a Rule**

To locate a rule by date, use the **Date Modified** filter. Here are details:
- Each rule’s **Date Modified** value indicates when the rule was created or most recently modified.
- After you select the filter, click **** to open the date selector.
- To specify a date, double-click that date.
- To specify a range of dates, single-click each date.
- In the date selector, the date or dates that you select are shaded. The current date is outlined.
- To clear the date filter, select a different value in the **Filter by** drop-down list.
- As an alternative approach, you can click **** , add the **Date Created** column to the display, and sort that column.

To locate a rule by identity, use the **Modified By** filter or the **Principal** filter. Here are details:
- The **Modified By** filter is based on who created or last updated a rule. If you want to see the **Modified By** column, click **** , and add that column to the display.
The Principal filter is based on who a rule is assigned to.

Both fields contain 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.

The Principal field can contain the principal type value Authenticated Users.

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.

The sasapp principal is a service account that provides necessary bootstrap access.

As an alternative approach, you can click , add the Created By column to the display, and sort that column.

**Restrict Creation of Top-Level Folders**

Initially, any user can create a top-level folder. To enable only SAS Administrators to create top-level folders, complete the following steps on the Rules page in SAS Environment Manager.

1. Locate and modify the rule that enables all users to create folders at all levels.
   a. In the Filter by drop-down list, make sure Object URI is selected.
   b. In the text box, enter /folders/folders.
   c. In the list of rules, select the rule that grants Authenticated Users the Add, Read, and Create permissions for the following objectUri: /folders/folders.
   d. Click .
   e. In the Edit Rule window, adjust values as follows:

<table>
<thead>
<tr>
<th>Permissions</th>
<th>In the Create tag, click x to remove that permission from the rule.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Revise the text. For example: Users can retrieve folders.</td>
</tr>
</tbody>
</table>

   Note: SAS Administrators can still create folders because there is a predefined rule that grants them broad access.

   f. Click Save.

2. Restore users’ ability to create lower-level folders in authorized locations.
   a. In the list of rules, select the rule that grants Authenticated Users the Create permission for the following objectUri: /folders/folders?parentFolderUri=/folders/folders/*.

   **CAUTION!** This predefined rule includes an advanced, specialized property setting that you cannot modify in SAS Environment Manager. Do not delete, copy, or attempt to recreate this rule.

   b. Click .
   c. In the Edit Rule window, enable the rule as depicted in the following figure:

   ```
   Enabled: 
   ```

   d. Click Save.
3  To verify the results, sign out of SAS Environment Manager, and then sign back in without assuming your membership in SAS Administrators.

- Make sure that you cannot add a top-level folder.
- Make sure that you can add a lower-level folder beneath a folder for which you have sufficient access. See Permissions by Task on page 15.

Note: Limiting the ability to create top-level folders is an example of managing access to functionality. The preceding instructions affect access to endpoints in the folders service, not access to a particular folder. To learn about managing access to functionality, see SAS Viya Administration: Identity Management.
General Authorization: Concepts

Scope
General authorization manages access to the following resources:
- content, such as folders, reports, and plans
- functionality, such as applications, features, and services

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>A resource or set of resources. Example: folderA, reportA</td>
</tr>
<tr>
<td>Principal</td>
<td>The user, group, or construct to which a rule is assigned. Example: UserA, GroupA, Authenticated Users</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:
- an individual authenticated user or service account
- a user group (a custom group or a group in your authentication provider)
- Authenticated Users (the construct that represents all authenticated users)
- Everyone (the construct that represents all authenticated users and any anonymous users)
Note: In the current release, we recommend that you avoid assigning rules to the Everyone principal. Not all interfaces enable you to introduce the Everyone principal.

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 about predefined groups, see 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, plans, 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 in SAS Environment Manager, 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 in SAS Environment Manager, 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>
</tbody>
</table>
### Permission Settings

<table>
<thead>
<tr>
<th>Permission</th>
<th>Affected Activity</th>
</tr>
</thead>
<tbody>
<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>Remove an object from a container.</td>
</tr>
</tbody>
</table>

* Applicable for a service, service endpoint, or media type.
** Applicable for a container, such as a folder.

### 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 information about managing access to functionality, see 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:
### 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), reports, and plans 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, Update</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.

Initial access to a new lower-level folder, report, or plan is determined by inheritance and any other influencing rules. A new lower-level folder, report, or plan does not have any automatically generated direct settings.

### 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.

**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.

**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**

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

A uniform resource identifier (URI) references a resource: a service, service endpoint, or particular object instance within a service. A URI does not correspond to a folder path or include a user-defined object name.

**URI Suffix**  
An objectUri that includes the /** suffix affects access to all descendant functionality. Here are examples:

- In an objectUri that references a service, the /** suffix affects access to the service and all of its endpoints (sub-functions within the service).
- In an objectUri that references a particular object within a service, the /** suffix provides access to the associated service’s endpoints in the context of that particular object.

In general, rules that specify the objectUri for a content object (such as a folder, report, or plan) should include the /** suffix. Inadvertently omitting the suffix narrows the effects of a rule and can yield unintended results due to insufficient access.
Note: The /** suffix references descendant functionality (service endpoints), not descendant content objects (such as reports within a folder). The /** suffix has nothing to do with container-based object inheritance. You cannot append the /** suffix to a containerUri.

In SAS Environment Manager, a suffix is included as follows:

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

Note: A specialized rule might use the suffix /*, which affects access to only immediate descendant functionality.

**objectUri**

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

- A rule that specifies a folder’s objectUri affects access to that folder.
- A rule that specifies a report’s objectUri affects access to that report.
- A rule that specifies a service’s objectUri, and does not specify a specific object instance, affects general access to functionality. See SAS Viya Administration: Identity Management.

**containerUri**

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

**mediaType**

A rule that specifies a mediaType applies to all instances of that media type.

### Rule Attributes

<table>
<thead>
<tr>
<th>Target-related attributes:</th>
<th></th>
</tr>
</thead>
</table>
| **objectUri**             | A relative URI for an object.
|                          | Character limit: 500 |
| **containerUri**          | A relative URI for 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. See Inheritance.
|                          | Character limit: 500 |
| **mediaType**             | A type of object, such as report. Rules that target a media type affect all object instances for that media type. Most rules do not specify a media type.
|                          | Character limit: 100 |

<table>
<thead>
<tr>
<th>Principal-related attributes:</th>
<th></th>
</tr>
</thead>
</table>
**principalType**  
Two of the values (Authenticated Users and Everyone) represent classes of users.  
Everyone includes all authenticated users and any anonymous users.  
- Assign broad grant rules to Authenticated Users.  
- Do not assign prohibit rules to Everyone or to Authenticated Users. Such rules block access for all users, including yourself.

**principal**  
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. See Principals.  
Character limit: 100

<table>
<thead>
<tr>
<th>Access-related attributes:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>type</strong></td>
<td>The indication of whether a rule blocks (prohibit) or attempts to provide (grant) access. Prohibit rules have absolute precedence.</td>
</tr>
</tbody>
</table>
| **condition**             | An expression that limits the scope or applicability of a rule. See Rule Conditions.  
Character limit: 5120 |
| **permissions**           | A list of access types. At least one permission is required. See Permissions. |
| **enabled**               | The indication of whether a rule is enabled. By default, rules are enabled. To temporarily prevent a rule from being enforced, disable the rule. |

<table>
<thead>
<tr>
<th>Documentation-related attributes:</th>
<th></th>
</tr>
</thead>
</table>
| **description**                  | Text that documents a rule for administrative purposes.  
Character limit: 1000 |
| **reason**                       | 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 |

**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)](https://docs.spring.io/spring-framework/docs/current/javadoc-api/org/springframework/context/expressions/SpELExpression.html).
- 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

#### 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, <code>en_US</code>).</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>
### 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, <code>application/vnd.sas.credential.domain+json</code>).</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>

### 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>LocalTime</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.ZoneId`.

### 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>method()</td>
<td>Method that the request invoked (for example, <code>GET</code>).</td>
<td>String</td>
</tr>
<tr>
<td>protocol()</td>
<td>Protocol of the request (for example, <code>HTTP/1.1</code>).</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):

\[
\text{localDate('US/Eastern').dayOfWeek} \neq \text{T(java.time.DayOfWeek).SUNDAY} \quad \text{and} \quad \text{localDate('US/Eastern').dayOfWeek} \neq \text{T(java.time.DayOfWeek).SATURDAY}
\]

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

\[
\text{#userId} == \text{currentUser()}
\]
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>All Relevant Rules</td>
<td>Effective Access and Explanation</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Grant + Conditional Grants</td>
<td>![Checkmark] Authorized. Relevant grants provide cumulative access, if there are no relevant prohibit settings.</td>
</tr>
<tr>
<td>Grant + Conditional Prohibits</td>
<td>![Flag] 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>![Flag] Conditional. Authorized for requests that are inside any of the grant conditions.</td>
</tr>
<tr>
<td>Conditional Grants + Conditional Prohibits</td>
<td>![Flag] 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*. 
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.
- Consider performing a backup before and after you make significant changes to your system. See SAS Viya Administration: Backup and Recovery.
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 in SAS Environment Manager, right-click a rule to view all of its properties. Or, add the Enabled column to the display, and sort that column by its values.

- Make sure you understand the precedence model. See Authorization Decisions on page 22.

- 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 on page 14.

- 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” in SAS Viya Administration: Orientation to Authorization.

- If the unexpected outcome is for access to multiple, disparate resources, use the Rules page to make sure a target-less rule was not accidentally introduced. Add the Container URI and Media Type columns to the display, and sort those columns to identify any rules that do not specify at least one target-related attribute. See Rule Targets on page 17.

  **TIP** If you know when a change occurred, or who made a change, see Tips for Locating a Rule on page 10.

- If the unexpected outcome is for access to a caslib or table, see SAS Viya Administration: Cloud Analytic Services Authorization.

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 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 in SAS Environment Manager:

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:
   - If the resource is a content object (such as a folder, report, or plan) and you cannot see the resource on the **Content** page, you lack Read access to the resource. Use the **Rules** page.
   - 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.
   - If the resource is not a content object, use the **Rules** page.
   - 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 do not know who introduced a problematic rule or when a problematic rule was introduced, right-click the **Setting** column on the **Rules** page, and select **Sort** so you can scan the prohibit rules.
   - If you cannot find a relevant prohibit rule, make sure that there is a relevant grant rule that provides access. See **Authorization Decisions on page 22**.

   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 the general authorization system.
   - **CAUTION! When the authorization system is disabled, authorization requirements are not enforced.**
   - 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 **SAS Viya Administration: Configuration Properties**.

   b. Restart the authorization service.
      
      Note: To learn how to restart services, see **SAS Viya Administration: General Servers and Services**.

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

6. Enable the general authorization system.
   - 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.
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.

*Interfaces to General Authorization*

<table>
<thead>
<tr>
<th>🌐 Authorization window in SAS Environment Manager.</th>
<th>The basic enterprise graphical interface for managing access to content such as folders, reports, and plans.</th>
</tr>
</thead>
<tbody>
<tr>
<td>⚖️ Rules page in SAS Environment Manager.</td>
<td>The advanced enterprise graphical interface for managing rules directly.</td>
</tr>
</tbody>
</table>