# SAS® Cloud Analytic Services 3.1: Authorization

## CAS Authorization: Overview

- Introduction .............................................................................................................. 3
- Navigation .................................................................................................................. 4
- Examine Access to a Caslib ....................................................................................... 4
- Provide Public Access to a Caslib .............................................................................. 5
- Selectively Share Access to a Caslib .......................................................................... 5
- Selectively Limit Access to a Caslib ........................................................................... 6
- Resolve Duplicate Access Controls .......................................................................... 7
- Manage Access to a CAS Server ................................................................................. 7

## CAS Authorization: How To (CAS Server Monitor)

- Introduction .............................................................................................................. 4
- Navigation .................................................................................................................. 4
- Examine Access to a Caslib ....................................................................................... 4
- Provide Public Access to a Caslib .............................................................................. 5
- Selectively Share Access to a Caslib .......................................................................... 5
- Selectively Limit Access to a Caslib ........................................................................... 6
- Resolve Duplicate Access Controls .......................................................................... 7
- Manage Access to a CAS Server ................................................................................. 7

## CAS Authorization: Concepts

- Scope ......................................................................................................................... 8
- Key Terms ................................................................................................................... 8
- Principals .................................................................................................................... 9
- Administrators .......................................................................................................... 9
- Inheritance ................................................................................................................ 9
- Permissions ............................................................................................................... 9
- Permissions by Task .................................................................................................. 10
- Row-Level Access .................................................................................................... 11
- Column-Level Access .............................................................................................. 14
- Access to Actions ...................................................................................................... 14
- Authorization Decisions ............................................................................................ 15
- Explanations of Effective Access ............................................................................. 15

## CAS Authorization: Guidelines

..................................................................................................................... 17

## CAS Authorization: Troubleshooting

..................................................................................................................... 18
CAS Authorization: Overview

To learn about the Cloud Analytic Services (CAS) authorization system, see CAS Authorization: Concepts.

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

- To interactively manage access to caslibs, use CAS Server Monitor. See CAS Authorization: How To (CAS Server Monitor) on page 4.
- To programmatically manage access to caslibs, tables, columns, and rows, use the Access Control action set. See SAS Cloud Analytic Services: System Programming Guide.
# CAS Authorization: How To (CAS Server Monitor)

## Introduction

These instructions explain how to manage access to global caslibs using CAS Server Monitor.

**Note:** For some items in the interface, CAS Server Monitor uses display names. This document uses technical names, except in specific references to interface items that have display names. If a display name differs from a technical name by more than capitalization and spacing, this document appends the technical name for clarity. For example: “Select **Promote Table** (Promote).”

**TIP** To manage access at the table, column, or row level, you must use another interface.

## Navigation

1. In CAS Server Monitor, beneath the **SAS Cloud Analytic Services** banner, click 💾.
2. On the Configuration page, select **Access Controls**.
3. In the **Caslibs** list, select the caslib.

Here are details:

- All of the global caslibs that you are authorized to see are listed.
- To see any new global caslibs, click 🔄. The **Caslibs** list is not updated automatically.
- Session and personal caslibs are not listed because you cannot set access controls on them. They are intrinsically private.
- The **Global Caslib Creation** and **Session Caslib Creation** caslibs do not contain data. These special caslibs determine which non-administrators can add and delete caslibs. See the information about caslib management privileges in **SAS Viya Administration: SAS Cloud Analytic Services**.

## Examine Access to a Caslib

On the Configuration page, under **Access Controls**, the columns for the selected caslib are populated as follows:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applies To</td>
<td>Specifies a type of access control principal.</td>
</tr>
<tr>
<td>Identity</td>
<td>Specifies the name for an access control principal.</td>
</tr>
<tr>
<td>Grant</td>
<td>A check mark indicates that access to the selected caslib is authorized for the specified principal and permission.</td>
</tr>
<tr>
<td>Deny</td>
<td>A check mark indicates that access to the selected caslib is not authorized for the specified principal and permission.</td>
</tr>
<tr>
<td>Activity</td>
<td>Specifies a permission, such as Select.</td>
</tr>
</tbody>
</table>
The rows for a new custom caslib are initially populated as follows:

- For the user who added the caslib, there is initially a row for every permission, because that user has a direct setting for each permission.
- For Authenticated Users, there is always a row for every permission, because Authenticated Users has an inherited setting for each permission.
- For other identities, there is a row for each direct setting. For example, if UserA has a direct grant of the Select permission, there is a Select row for UserA.

**Provide Public Access to a Caslib**

To give all users access to a new global caslib that you added:

1. On the Configuration page, select Access Controls.
2. In the Caslibs pane, select the caslib.
3. Click Edit.
4. In the Edit Access Controls window, adjust settings as follows:
   a. In the Authenticated Users row for Read Info, select the Grant radio button.
   b. Click Add Row. In the new row at the end of the page, select Authenticated Users, the Grant radio button, and the Select activity.
   c. If you want to also provide Write access, add rows that grant the following additional permissions to Authenticated Users: Insert, Update, Delete, Create Table, Drop Table, Delete Source, Alter Table, Limited Promote, Promote Table (Promote).

   **TIP** As an alternative to adding each row individually, click Add Set, and select Add Set from the drop-down list. In the new Authenticated Users rows for Alter Caslib and Manage Access, click to delete those direct access controls. In the remaining new rows, select the Grant radio button.

5. Click OK to save your changes.
6. Under Access Controls, review the results of your changes.

**Selectively Share Access to a Caslib**

This example gives UserA Read and Write access to a new global caslib that you added:

1. On the Configuration page, under Access Controls, select the caslib.
2. Click Edit.
3. In the Edit Access Controls window, click Add Set, and select Add User Set from the drop-down list.
4. In the next window, enter UserA in the User name field. Click OK.
5. In the Edit Access Controls window, select the Grant radio button in each UserA row, except the Alter CASLib and Manage Access rows.
6. At the end of UserA's Alter CASLib and Manage Access rows, click .
7. Click OK to save your changes.
Under **Access Controls**, review the results of your changes.

Note: UserA does not have rows for **Alter CASLib** and **Manage Access**, because you did not give UserA direct access controls for those permissions. Unless UserA is a member of a group that has direct access controls for those permissions, UserA’s effective access for those permissions comes from Authenticated Users.

Here are some additional details about editing access controls:

- To add a single row, click **Add Row**.
- To provide a complete set of editable rows for an identity, click **Add Set** and then select the appropriate item:
  - For a group, select **Add Group Set**.
  - For a user, select **Add User Set**.
  - For Authenticated Users, select **Add Set**.
- To delete a direct access control, click **删除**.
- You cannot change or delete inherited settings. You can add a direct access control that has precedence over an inherited setting.

**Selectively Limit Access to a Caslib**

**About Setting Direct Denials**

**CAUTION!** Identity names that you enter are not validated. For sensitive data, do not grant access to Authenticated Users and then rely on selective direct denials. The safer practice is to verify that access is broadly denied, and then grant access selectively.

**CAUTION! Do not block your own access.** Before you add a direct denial for a group that you belong to, make sure you have a higher precedence (offsetting) direct grant. If you are an administrator who is exempt from data authorization requirements, this precaution is not strictly necessary.

**Block All Access for an Identity**

To block all access for an identity:

1. On the **Configuration** page, under **Access Controls**, select the caslib.
2. Click Edit.
   - **Note:** If the Edit button is disabled, you are not authorized to set permissions for the selected caslib.
3. In the **Edit Access Controls** window, click **Add Row**.
4. In the new row, select an identity type, enter a name, make sure the Deny radio button is selected, and select the Read Info activity.
5. Click OK to save your changes.
6. Under **Access Controls**, review the results of your changes.

**Block Write Access for an Identity**

To block Write access for an identity:

1. On the **Configuration** page, under **Access Controls**, select the caslib.
2 Click **Edit**.

*Note:* If the **Edit** button is disabled, you are not authorized to set permissions for the selected caslib.

3 In the **Edit Access Controls** window, click **Add Set**, and select **Add Group Set** or **Add User Set**.

4 In the next window, enter the user or group name. Click **OK**.

5 In the identity’s **Read Info** and **Select** rows, click **X**.

6 In the identity’s remaining rows, make sure the **Deny** radio button is selected.

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

8 Under **Access Controls**, review the results of your changes.

---

**Resolve Duplicate Access Controls**

You cannot save more than one direct setting for a particular caslib, principal, and permission. Here are examples:

- You cannot save two direct grants of the Update permission for UserA on caslibA.
- You cannot save both a direct denial and a direct grant of the ReadInfo permission for UserA on caslibA.

If there is one duplicate direct setting in the **Edit Access Controls** window when you click **OK**, the error message **Access control has duplicates** is displayed twice. If you delete either of the settings that has the error message, one error message remains. At that point, you can save your changes by clicking **OK** again.

---

**Manage Access to a CAS Server**

You cannot set access controls on a CAS server. CAS authorization begins at the caslib level. Access to a CAS server is determined by authentication. See *SAS Viya Administration: Authentication*.

See *SAS Viya Administration: SAS Cloud Analytic Services* for information about caslib management privileges and available paths.

See *SAS Viya Administration: Identity Management* for information about CAS roles.
CAS Authorization: Concepts

Scope

Authorization is the aspect of security that determines which resources are available to which users. CAS authorization manages access to the following CAS objects:
- caslibs
- CAS tables and columns
- CAS action sets and actions

CAS authorization requirements do not apply in the following circumstances:
- The requesting user has assumed a role that is exempt from authorization requirements. For example, the user has assumed the Superuser role.
- The target object is intrinsically private. For example, the target is a table in a personal caslib, a session caslib, or the session scope of a global caslib.

Note: This topic explains the entire CAS authorization model. Not all interfaces expose all aspects of the model.

Note: To learn about CAS, see SAS Cloud Analytic Services: Fundamentals.

Key Terms

<table>
<thead>
<tr>
<th>Access control</th>
<th>A composite of authorization elements. Example: An access control grants ReadInfo to groupA on caslibA.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td>A resource. Examples: tableA, caslibA</td>
</tr>
<tr>
<td>Principal</td>
<td>The user, group, or construct to which an access control is assigned. Examples: UserA, GroupA, Authenticated Users</td>
</tr>
<tr>
<td>Permission</td>
<td>A type of access. Values: ReadInfo, Select, LimitedPromote, Promote, CreateTable, DropTable, DeleteSource, Insert, Update, Delete, AlterTable, AlterCaslib, ManageAccess</td>
</tr>
<tr>
<td>Setting</td>
<td>An indication of whether (and to what extent) access is provided. Values: Grant, Row-Level Grant, Deny</td>
</tr>
<tr>
<td>Filter</td>
<td>In a row-level grant of the Select permission, the constraint expression. Example: User='SUB::SAS.Userid', sales&gt;1000</td>
</tr>
<tr>
<td>Effective access</td>
<td>A context-neutral description of the net result of all relevant access controls. Values: Authorized, Not Authorized, Row-Level</td>
</tr>
<tr>
<td>Access outcome</td>
<td>In an access request, the authorization decision. Values: Authorized, Not Authorized, Row-Level Authorization</td>
</tr>
</tbody>
</table>
Principals
The principal in an access control is the user, group, or construct to which the access control is assigned. The CAS authorization system supports the following principals:
- an individual authenticated user
- a user group
- Authenticated Users (the construct that represents all authenticated users)
  Note: In some contexts, this construct corresponds to the group that is named *.

Administrators
CAS roles provide per-server assumable access to administrative functionality. For example, the Superuser role provides unrestricted access to data and actions. See SAS Viya Administration: Identity Management.

Inheritance
Access flows through a hierarchy of objects. Each parent object conveys settings to its child objects. Each child object inherits settings from its parent object.
Here are the inheritance relationships:
- Access flows from a caslib to its tables.
- Access flows from a table to its columns.
- Access flows from an action set to its actions.
Note: Each caslib has a default access value that provides inherited settings for that caslib. In the current release, the default access value is Deny, which yields inherited denials of all permissions for the Authenticated Users group (group *).

Permissions

<table>
<thead>
<tr>
<th>Permission</th>
<th>Data Enforcement Levels</th>
<th>Affected Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Caslib</td>
<td>Table</td>
</tr>
<tr>
<td>ManageAccess</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ReadInfo</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>LimitedPromote</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Promote</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>CreateTable</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>DeleteSource</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>DropTable</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Select</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Permission</td>
<td>Data Enforcement Levels</td>
<td>Affected Activities</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------</td>
<td>--------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AlterCaslib</td>
<td>✓</td>
<td>Change the properties of a caslib.</td>
</tr>
<tr>
<td>AlterTable</td>
<td>✓</td>
<td>Change the attributes or structure of a table.</td>
</tr>
<tr>
<td>Insert</td>
<td>✓</td>
<td>Add rows.</td>
</tr>
<tr>
<td>Delete</td>
<td>✓</td>
<td>Delete rows.</td>
</tr>
<tr>
<td>Update</td>
<td>✓</td>
<td>Change data values.</td>
</tr>
<tr>
<td>Execute</td>
<td></td>
<td>Run an action.</td>
</tr>
<tr>
<td>Load</td>
<td></td>
<td>Load an action set.</td>
</tr>
</tbody>
</table>

* You can set permissions at or above the level where they are enforced. See also [Access to Actions](#).

** To delete any direct access controls, the ManageAccess permission is required.

## Permissions by Task

To provide sufficient access to complete a task, you must consider access to all relevant objects. The following tables document the permissions that are required for each data object when performing selected tasks.

### Simple Tasks

<table>
<thead>
<tr>
<th>Task (CAS Action)</th>
<th>Caslib</th>
<th>Table</th>
<th>Column</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modify caslib properties</td>
<td>ReadInfo</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>AlterCaslib</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set table permissions</td>
<td>ReadInfo</td>
<td>ReadInfo</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>ManageAccess</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Load a table from a caslib’s data source (loadTable)</td>
<td>ReadInfo</td>
<td>ReadInfo</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Select</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transfer and load an entire file (upload)</td>
<td>ReadInfo</td>
<td>ReadInfo</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Select</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transfer rows to the server (addTable)</td>
<td>ReadInfo</td>
<td>ReadInfo</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Select</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Move a table to global scope (promote)</td>
<td>ReadInfo</td>
<td>ReadInfo</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Promote</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remove a table from global scope (dropTable)</td>
<td>ReadInfo</td>
<td>ReadInfo</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>DropTable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delete a file (deleteSource)</td>
<td>ReadInfo</td>
<td>ReadInfo</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>DeleteSource</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Persist a file (save) | ReadInfo | ReadInfo | -
| | CreateTable | CreateTable |
| | | | (DeleteSource)

Read data | ReadInfo | ReadInfo | ReadInfo |
| | | Select | Select |

Insert rows | ReadInfo | ReadInfo | -
| | | Insert | |

Update rows | ReadInfo | ReadInfo | -
| | | Select | Update |

* For promotion of a table within the same caslib, LimitedPromote for the table (instead of Promote for the caslib) is sufficient.

** DeleteSource is required to replace a source table.

**Caslib Management Tasks**

<table>
<thead>
<tr>
<th>Task</th>
<th>Who Can Perform the Activity</th>
</tr>
</thead>
</table>
| Add global caslibs | Superusers and Data administrators  
Users who have global caslib management privileges |
| Add session caslibs | Superusers and Data administrators  
Users who have session caslib management privileges |
| Delete global caslibs | Superusers and Data administrators  
Users who have global caslib management privileges can delete any global caslib for which they have the ReadInfo and ManageAccess permissions. |
| Delete session caslibs | Superusers and Data administrators  
Users who have session caslib management privileges can delete any session caslib for which they have the ReadInfo and ManageAccess permissions. |
| Adjust caslib management privileges | Superusers and Data administrators |

**Note:** Global caslib management privileges correspond to the ManageAccess permission on the _GLOBAL caslib. Session caslib management privileges correspond to the ManageAccess permission on the _SESSION caslib. See *SAS Viya Administration: SAS Cloud Analytic Services*.

**Row-Level Access**

**Introduction**

A row-level grant includes a filter that limits the Select permission on a table. A user who has row-level access to a table can view only those rows that are within the associated filter.
For example, you can use a row-level grant to enable groupA to see only those rows in tableA where the value in the Toy_Price column is 25. Here is an overview of the process:

1. On tableA, give groupA a row-level grant of the Select permission.
   
   Specify the following filter: Toy_Price=25

2. Make sure that groupA has ReadInfo access to tableA and its parent caslib.

3. Make sure that groupA is not a member of another group that has a grant or denial of the Select permission on tableA.

4. Verify that when a member of groupA accesses tableA, the expected rows are returned.

### Syntax for Row-Level Filters

<table>
<thead>
<tr>
<th>Operator (Alias)</th>
<th>Example Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contains (?)</td>
<td>Toy_Type Contains 'cars'</td>
</tr>
<tr>
<td>Not Contains</td>
<td>Toy_Type Not Contains 'cars'</td>
</tr>
<tr>
<td>In</td>
<td>Toy_Type In ('dolls' 'cars' 'animals')</td>
</tr>
<tr>
<td>Not In</td>
<td>Toy_Type Not In ('dolls' 'cars' 'animals')</td>
</tr>
<tr>
<td>Between -inclusive</td>
<td>Toy_Price Between 20 AND 30</td>
</tr>
<tr>
<td>Not Between -inclusive</td>
<td>Toy_Price Not Between 20 AND 30</td>
</tr>
<tr>
<td>Like</td>
<td>Toy_Type Like 'd%'</td>
</tr>
<tr>
<td>= (EQ)</td>
<td>Toy_Price=25</td>
</tr>
<tr>
<td>&gt; (GT)</td>
<td>Profit &gt; (Sales * .5)</td>
</tr>
<tr>
<td>&lt; (LT)</td>
<td></td>
</tr>
<tr>
<td>&gt;= (GE)</td>
<td></td>
</tr>
<tr>
<td>&lt;= (LE)</td>
<td></td>
</tr>
<tr>
<td>^= (NE, ~=)</td>
<td></td>
</tr>
<tr>
<td>+ -addition</td>
<td></td>
</tr>
<tr>
<td>- -subtraction</td>
<td></td>
</tr>
<tr>
<td>/ -division</td>
<td></td>
</tr>
<tr>
<td>* -multiplication</td>
<td></td>
</tr>
<tr>
<td>** -exponentiation</td>
<td></td>
</tr>
<tr>
<td>() -parentheses</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>AND (&amp;)</td>
<td>Toy_Type='cars' OR Toy_Type='dolls'</td>
</tr>
<tr>
<td>OR (</td>
<td>, !)</td>
</tr>
<tr>
<td>NOT</td>
<td></td>
</tr>
</tbody>
</table>
### Operator (Alias) Example Filter

<table>
<thead>
<tr>
<th>Operator (Alias)</th>
<th>Example Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is Missing</td>
<td>Toy_Type Is Not Null</td>
</tr>
<tr>
<td>Is Not Missing</td>
<td></td>
</tr>
<tr>
<td>Is Null</td>
<td></td>
</tr>
<tr>
<td>Is Not Null</td>
<td></td>
</tr>
</tbody>
</table>

### Identity-Based Substitution

Identity-based substitution is a powerful and concise technique for defining row-level access. You can use substitution to implement any number of per-user access distinctions with a single row-level filter.

Identity-based substitution parameters map a user’s authenticated ID or group memberships to values in a specified column in your data. Values are dynamically substituted into the filter at run time, as appropriate for each requesting user. Here are the supported substitution parameters:

<table>
<thead>
<tr>
<th>Substitution Parameter</th>
<th>Example Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUB::SAS.Userid</td>
<td>Determines whether a data value is the same as the requesting user’s authenticated ID.</td>
</tr>
<tr>
<td></td>
<td>empID = 'SUB::SAS.Userid'</td>
</tr>
<tr>
<td>SUB::SAS.IdentityGroups</td>
<td>Determines whether a data value matches any of the requesting user’s group memberships.</td>
</tr>
<tr>
<td></td>
<td>FacilityRegion In ('SUB::SAS.IdentityGroups')</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The comparison is against each group’s unique name, so your data must contain unique group names.</td>
</tr>
</tbody>
</table>

### Example: User ID Substitution

If a tableB has an empID column with values that match the user IDs with which users authenticate, you might assign this filter to Authenticated Users:

empID = 'SUB::SAS.Userid'

At request time, each user’s ID is substituted into the right side of the expression. In a request from userA, the expression resolves as:

empID = 'userA'

As a result, userA gets only those rows where the value in the empID column is userA.

### Example: Membership Substitution

If tableC has a FacilityRegion column with values that match the unique names for user groups, you might assign this filter to an AllRegions group:

FacilityRegion In ('SUB::SAS.IdentityGroups')

At request time, each affected user’s list of group memberships is substituted into the right side of the expression. In a request from user13 (who is a member of the grp7, grp9, and AllRegions groups), the expression resolves as:

FacilityRegion In ('grp7', 'grp9', 'AllRegions')
As a result, user13 gets only those rows where the value in the FacilityRegion column is *grp7, grp9,* or *AllRegions.*

**Note:** *Authenticated Users* is not one of the listed memberships, because it is an access control principal, not a user group.

**Multiple Filters and Cumulative Access**

If multiple row-level filters are applicable to a user, only the highest precedence filter provides access. If there is an identity precedence tie (the user is a member of multiple groups, each of which has a filter), the user can access any row that meets any of the filters.

Here are details:

The filters for multiple row-level grants provide cumulative access only if all of the following circumstances exist:

- The requesting user does not have a direct access control for the Select permission.
- None of the requesting user’s groups have a direct grant or denial for the Select permission.
- Two or more of the requesting user’s groups have row-level grants.

**Note:** All explicit groups have equal precedence, regardless of any nested memberships.

**Note:** A filter for a row-level grant that is assigned to Authenticated Users is never cumulative (joined with other filters by OR). Authenticated Users is a construct that has lower precedence than any group.

**Column-Level Access**

CAS supports column-level access distinctions. However, interface-specific limitations apply as follows:

- In SAS Visual Analytics reports, column-level access distinctions are not supported. For a user who lacks access to any column in a table, no data for that table is available in a SAS Visual Analytics report.
- You cannot use a graphical interface to set access controls on columns.

**Note:** You can set column-level permissions using the action set *interface.*

**Access to Actions**

Action sets and actions that have no access controls are available to all users. As a result, almost all action sets and actions are available to all users. In general, the ability to perform a particular task is managed by access controls on the target data, not by access controls on actions.

An exception is actions for adding nodes and stopping the server. The initial configuration denies Authenticated Users the Execute permission for those actions. Initially, only Superusers can add nodes or stop the server.

Here are additional details:

- The ReadInfo and Execute permissions are enforced for actions.
- The ReadInfo and Load permissions are enforced for action sets.
- Unregistered action sets are subject to access constraints that are defined on the _UNREGISTERED action set. The initial configuration denies Authenticated Users the Load permission on the _UNREGISTERED action set.

**Note:** An unregistered action set is an action set that is not listed in the database of action sets that SAS provides. SAS solutions use only registered action sets.

**Note:** An attempt to load an action set that does not exist generates an access denied error message, because no such action set is known and registered. For example, if you do not correctly specify the action set name in a load request, an access denied error message is generated.
Authorization Decisions

Precedence
In the CAS authorization system, precedence is determined by where an access control is set and who an access control is assigned to.

Direct access controls have precedence over inherited access controls, regardless of who the principal is. For example, if only the following access controls exist, then UserA cannot access TableA:

- UserA has a direct grant of ReadInfo on caslibA.
- Authenticated Users has a direct denial of ReadInfo on TableA, which is in caslibA.

Note: One way to enable UserA to access TableA is to add a direct grant of ReadInfo for UserA on TableA. UserA's direct grant has precedence over the direct denial for Authenticated Users.

The principal precedence hierarchy is relatively flat. It consists of only the following three levels:

1. An individual user.
2. A group. All group memberships are at the same level of precedence, even if groups are nested.
3. Authenticated Users.

How Access Is Evaluated
Each access request initiates an authorization decision process. That process terminates when an outcome is reached. Here is the authorization decision process for a data access request:

1. If the requesting user has assumed the Superuser or Data role, they are exempt from data authorization requirements. The outcome is Authorized.

2. If there are relevant direct access controls on the target object, those access controls determine the outcome as follows:
   a. If there is a setting that is specifically assigned to the requesting user, that setting determines the outcome.
   b. If there is a denial from a group, the outcome is Not Authorized.
   c. If there is a grant from a group, the outcome is Authorized.
   d. If there is exactly one row-level grant from a group, the outcome is Row-Level Authorization (authorized for rows within the applicable filter).
   e. If there are two or more row-level grants from groups, the outcome is Row-Level Authorization (authorized for any row that is within any of the applicable filters). See Multiple Filters and Cumulative Access.
   f. If there is a setting for Authenticated Users, that setting determines the outcome.

3. Otherwise, the outcome is determined by the caslib’s default value. By default, that value is Deny, so the outcome is Not Authorized.

Explanations of Effective Access
Origins information explains effective access by answering the question: Why does this identity have this effective access to this object?
Origins information identifies the highest precedence access control that causes the access outcome for a particular identity, object, and permission. If there are multiple tied highest precedence access controls, origins information includes all of them. Additional, lower precedence controls are not included.

The following table provides simple examples of origins information. Each row in the table is for a different (independent) scenario. In each example, we are looking at why UserA has an effective access value of Not Authorized for the ReadInfo permission on TableA. UserA is a member of GroupA and GroupB. TableA is in CaslibA.

**Origins: Examples**

<table>
<thead>
<tr>
<th>Highest-Precedence Access Control (or Controls)</th>
<th>Origins Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>On TableA, a direct denial for UserA.</td>
<td>TableA UserA</td>
</tr>
<tr>
<td>On TableA, a direct denial for GroupA.</td>
<td>TableA GroupA</td>
</tr>
<tr>
<td>On TableA, direct denials for GroupA and GroupB.</td>
<td>TableA GroupA, GroupB</td>
</tr>
<tr>
<td>On TableA, a direct denial for Authenticated Users.</td>
<td>TableA Authenticated Users</td>
</tr>
<tr>
<td>On CaslibA, a direct denial for UserA.</td>
<td>CaslibA UserA</td>
</tr>
<tr>
<td>On CaslibA, a direct denial for GroupA.</td>
<td>CaslibA GroupA</td>
</tr>
<tr>
<td>On CaslibA, direct denials for GroupA and GroupB.</td>
<td>CaslibA GroupA, GroupB</td>
</tr>
<tr>
<td>On CaslibA, a direct denial for Authenticated Users.</td>
<td>CaslibA Authenticated Users</td>
</tr>
<tr>
<td>On CaslibA, an inherited denial for Authenticated Users.</td>
<td>Caslib default</td>
</tr>
</tbody>
</table>

**Note:** To obtain origins information, use the action whatIsEffective in the access controls action set. See *SAS Cloud Analytic Services: System Programming Guide.*
CAS Authorization: Guidelines

The following guidelines can contribute to simplicity and security:

- Limit membership in administrative roles.
- Minimize use of individual tables as targets.
- Minimize use of individual users as principals.
- Remember that any access that is not granted is implicitly denied. Do not set unnecessary denials.
- If you deny someone access to part of a table (using a column-level or row-level access control), make sure that identity cannot update or insert rows in that table.
- If you use a programmatic interface, see the details for the Access Control action set in SAS Cloud Analytic Services: System Programming Guide.
CAS Authorization: Troubleshooting

If you inadvertently block your own access to an object, contact an administrator for assistance.

Note: Anyone who has the ReadInfo and ManageAccess permissions for the object can reinstate your access.
CAS Authorization: Interfaces

All CAS authorization requirements and constraints are always fully enforced. However, not all interfaces enable you to see and interact with all CAS authorization features.

In the following table, the shaded part of each circle is an approximation of the amount of CAS authorization functionality that a particular interface exposes.

**Interfaces to CAS Authorization**

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAS Server Monitor</td>
<td>A graphical web application that is embedded in the CAS server. Supports managing access at the caslib level.</td>
</tr>
<tr>
<td>Access Control action set</td>
<td>A programmatic interface for CASL (the CAS procedure), Python, and Lua. See <a href="#">SAS Cloud Analytic Services: System Programming Guide</a>.</td>
</tr>
</tbody>
</table>