Teamcenter 13.3

Access Manager

Teamcenter 13.3
PLM00020 - 13.3

SIEMENS

About Siemens Digital Industries Software

Siemens Digital Industries Software is a leading global provider of product life cycle management (PLM) software and services with 7 million licensed seats and 71,000 customers worldwide. Headquartered in Plano, Texas, Siemens Digital Industries Software works collaboratively with companies to deliver open solutions that help them turn more ideas into successful products. For more information on Siemens Digital Industries Software products and services, visit **www.siemens.com/plm**.

Support Center: **support.sw.siemens.com**

Send Feedback on Documentation: **support.sw.siemens.com/doc_feedback_form**

Contents

Creating and managing access control lists (ACLs)

Distributing, reverting, and repairing the rule tree

Access Manager automated test harness

Verifying the effect of access rules

1. Getting started with Access Manager

Managing your users' access to data using Access Manager

Managing how users access your company data is an important factor in information security. Users may be employees within your company, or they may be external users such as suppliers and contractors.

Access Manager enables you to control user access to data objects stored in Teamcenter by:

- **Defining rules.**

- **Defining access control lists (ACLs).**

Rules and ACLs are used in combination with information about the user, such as group membership, project membership, nationality, and clearance level, which together determine the user's authorization to interact with data.

> Note:
>
> With the exception of the **Create** privilege, rules and ACLs do not control the creation of objects. They only determine what operations can be performed on existing objects. An administrator controls which objects a user can create using other means such as:
>
> - Using the **Create privilege** to block creation of certain objects
>
> - Using the Command Suppression application to suppress the display of menus and commands
>
> - Deploying a BMIDE condition to prevent creation of certain objects, which is commonly used in the Change Management module
>
> - Deploying a BMIDE type display rule to create display rules that hide specific types when creating new objects using the **File→New** menus

Access Manager is an administrative application that leverages:

- User information maintained in the Organization application.

- Project information created using the Project application.

- Object metadata and business rules that are defined and maintained using the Business Modeler IDE.

Before you begin

Prerequisites	You need Teamcenter administrator privileges to use the Access Manager application.
Enable Access Manager	Access Manager does not need to be enabled before you use it.
	If you have trouble accessing Access Manager, see your system administrator; it may be a licensing issue.

> **Note:**
>
> You can log on to Teamcenter only once. If you try to log on to more than one workstation at a time, you see an error message.

Configure Access Manager	Access Manager does not need to be configured.
Start Access Manager	Click **Access Manager** 🔒🔑 in the navigation pane.
View administration data	The Administration Data Report site located in the References for Administrators and Customizers contains the Administration Data Documentation report, which provides a list of default administration values.
	Select the **Access Manger** tile to expand the view of all default elements (rules, named ACLs, and privileges) with descriptions and values.
	Select the **Preferences** tile for information about the default preferences and their values.
	The Administration Data Report is described in more detail in the *Managing Administration Data* manual.

Access Manager interface

Access Manager interface overview
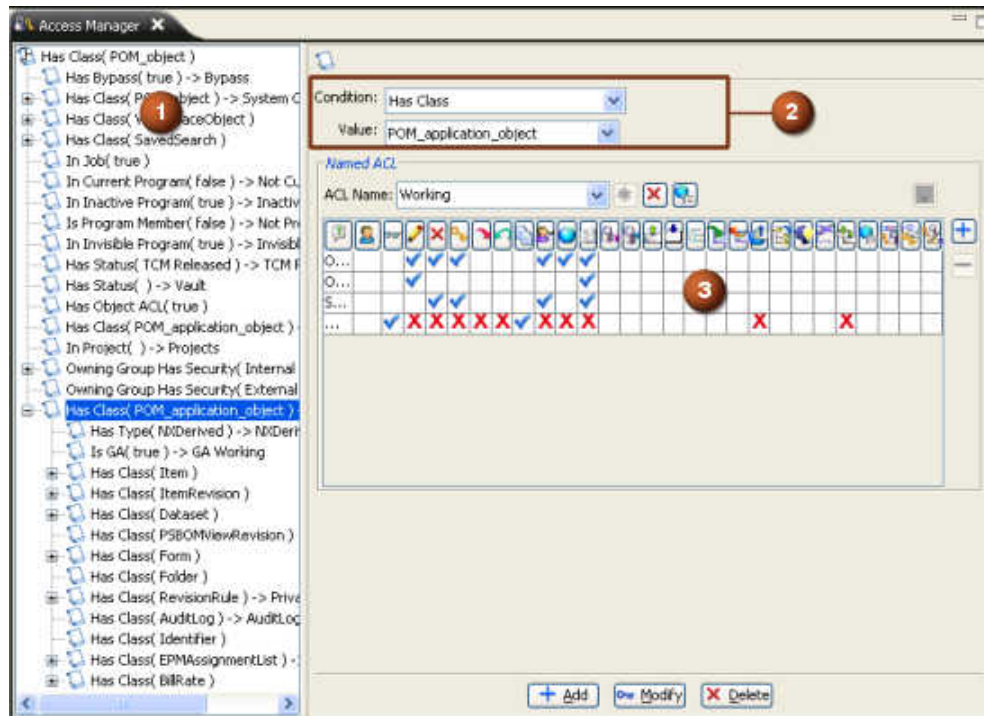
1	Rule tree pane	Enables you to view the structure of your access rules by expanding and collapsing branches. Select a rule in the tree to see the rule properties and named ACLs in the rule properties pane and the named ACL table.
2	Rule properties	Displays the condition and value for the rule selected in the rule tree. You can modify these properties and then create or modify a rule. You can delete the selected rule.
3	Named ACL table	Displays the ACL name and accessor entries for the rule selected in the rule tree. You can create, modify, and delete named ACLs.

Access Manager menus

Menu command	Description
File→Import	Browses for the ASCII file containing the rule tree data and then imports the file.
File→Export	Browses for the ASCII file containing the rule tree data and then exports the file.
Edit→Up	Moves a rule tree entry up one branch at a time within the same level.
Edit→Down	Moves a rule tree entry down one branch at a time within the same level.
View→Expand Below	Expands the rule tree to display subbranches.

Access Manager buttons

Button	Description
Move Rule Up	Moves a rule tree entry up one branch at a time within the same level.
Move Rule Down	Moves a rule tree entry down one branch at a time within the same level.
Add	There are two **Add** buttons: • The button to the right of the access control entry (ACE) table adds a new row to the table. • The button at the bottom of the pane adds the rule to the Access Manager tree.
Modify	Modifies the selected rule and/or access control list (ACL).
Delete	There are two **Delete** buttons: • The button to the right of the **ACL Name** box deletes the selected ACL. • The button at the bottom of the pane deletes the selected rule from the Access Manager tree.
Save	There are two **Save** buttons: • The button at the top right of the ACE table saves the ACL.

Button	Description
	• The button in the toolbar saves changes to the rule tree.
Create ACL ☀	Creates the ACL after you enter a name in the **ACL Name** box.
Localization 🌐	Displays the **Language Translations** dialog box that lists existing translation values for the names of ACL rules. By default, it is disabled. Enable it by selecting an ACL.

Access Manager symbols

Access Manager uses symbols to represent **privileges** that can be granted using access control lists (ACLs).

Basic concepts for using Access Manager

Protecting Teamcenter data

Object protection and ownership are extremely important in a distributed computing environment. Objects represent actual product information in the database and must be protected from unauthorized or accidental access, modification, and deletion. Teamcenter implements two different tiers of data protection:

- **Rules-based protection** is the primary security mechanism.

- **Object-based protection** is a secondary security mechanism that allows you to grant exceptions to rules.

Rules-based protection

Rules provide security for your Teamcenter data by:

- Controlling access to data on a global basis.

- Determining whether a user has permission to view or perform an action on an object.

- Filtering data according to the attributes of the data.

- Granting privileges to the data according to the users' IDs and their session context (the group and role they used to log on).

> **Note:**
>
> Rules do not control the creation of objects. They only determine what operations can be performed on existing objects.

Rules are defined by a combination of:

- A condition.

- A value for the condition.

- An access control list (ACL) that grants privileges to accessors.

The condition and value identify the set of objects to which the rule applies; the ACL defines the privileges granted to users (accessors).
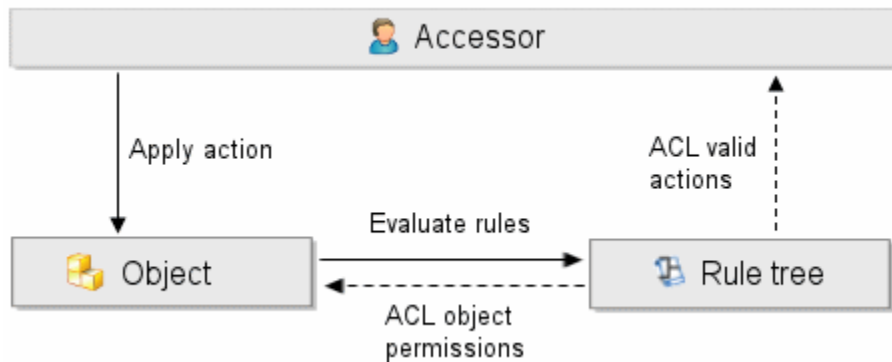
User actions against objects cause the rule tree to be evaluated to dynamically build an access control list for the object. The ACL controls permissions for the object and determines who (accessors) can do what (actions) to the object.

Object access control lists

Object-based protection uses access control lists (ACLs) to create exceptions to rules-based protection on an object-by-object basis.

Object ACLs are most useful when you need to:

- Grant wider access to a specific object.

- Limit access to a specific object.
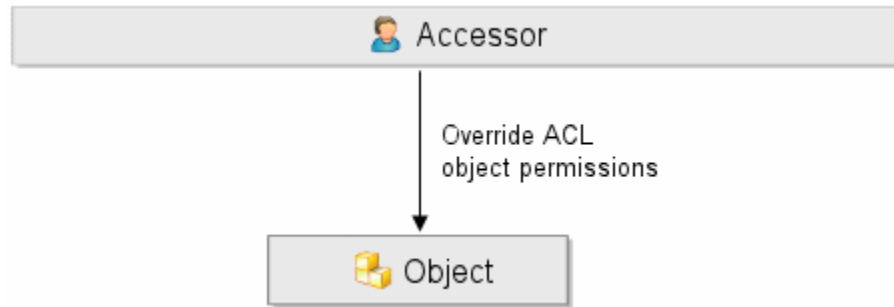
Teamcenter uses ACLs to determine access to an object. Users with proper permissions can override the ACL for an object to grant or deny permissions for certain users but only when the rule tree allows.

For example, the rule tree does not allow object-based access rules to override the rules-based protection when:

- An object has an assigned status.

- The object access rule is granted in a workflow.

> Note:
>
> ACLs do not control the creation of objects. They only determine what operations can be performed on existing objects.
>
> - Each ACL contains a list of accessors and the privileges granted, denied, or not set for each accessor.
>
> - Each individual pairing of an accessor with their privileges is considered a single access control entry (ACE).

Access control lists

Access control lists (ACLs) contain a list of accessors and the privileges granted, denied, or not set for each accessor. *Accessors* are collections of users who share certain common traits, such as membership in the group that owns the object or membership in the project team. Just as rules have a precedence weighting in the rule tree, **accessor precedence** weighting is considered when the ACL is evaluated.

Each pairing of an accessor with corresponding **privileges** in the list is referred to as an *access control entry (ACE)*. An ACL can be comprised of one or many ACEs.

ACLs are associated with conditions in the rule tree as part of a rules-based security model, and they can be used in more than one rule.

In addition, object ACLs grant exceptions to rules-based protection and are created by users with change privileges.

Access control lists display the current protections for an object.

> **Note:**
>
> - If an ACL is modified by a user, other users who are logged on at the same time are not affected by the updated ACL until they log off and log on again.
>
> - ACLs do not control the creation of objects. They only determine what operations can be performed on existing objects.

System Administrator		✔	✔				✔		✔
World	✔	✘	✘	✘	✘	✘	✔	✘	✘	✘

Lifecycle of data

All data in an enterprise typically passes through three basic phases, **Released**, **In-Process**, and **Working**.

Data state	Description
Released	Data is formalized and must be protected from modification. Released data is often consumed by users outside the authoring group; whereas, in-process and working data is consumed by authors and generally requires more restrictive read access.
In-Process	Data is semiformalized and because it is in the process of being released, it is assumed to be accurate and in its final form. However, allowances must be made for last-minute changes. The primary objective for protecting in-process data is to ensure that it is tightly controlled while it is being released.
Working	Data is not very firm and is expected to undergo many changes before it is released. The objective for protecting working data is to ensure that only the proper persons have permission to view, modify, or manipulate the data.

Access Manager rule tree

Rules are organized in the Access Manager rule tree and are evaluated based on their placement within the tree structure. The default rule tree included in your Teamcenter installation assumes that users are granted privileges unless explicitly denied.

The rule tree acts as a filter that an object passes through when a user attempts to access the object. When conditions that apply to the selected object are met, the privileges defined in the ACL are applied.

- The rules are evaluated from the top to the bottom of the tree.

- Rules at the top take precedence over rules at the bottom of the tree.

- Subbranches always take precedence over parent branches in the tree.

The rule tree appears to the left of the Access Manager window.
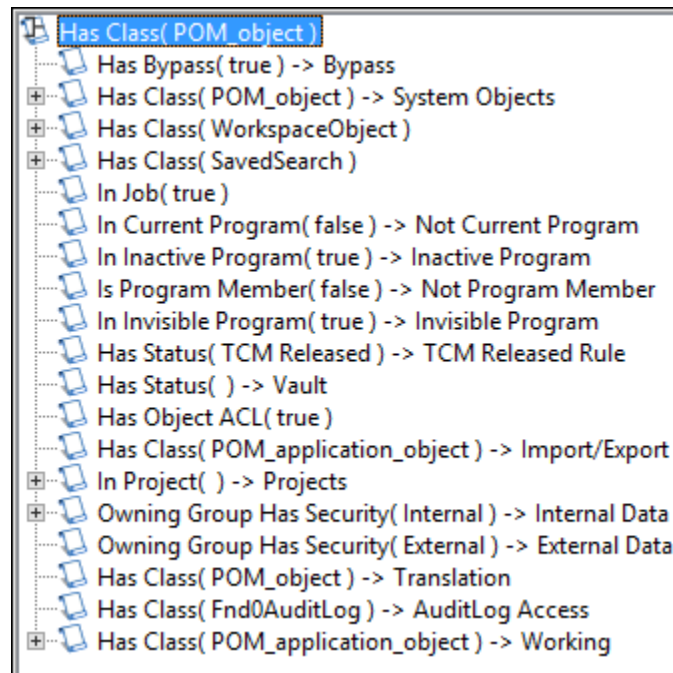
For a list of default rule conditions, see the *Access Manager Guide*.

Basic tasks using Access Manager

Using Access Manager, you can:

- **Create, modify, and delete rules**.

- **Create, modify, and delete access control lists (ACLs)**.

- **Export and import the rule tree**.

Upgrade Access Manager rules

Special steps are required to upgrade the Access Manager rule tree. These steps are required to ensure the rule tree in your upgraded system contains any new rules added by Teamcenter and also any custom rules you added to your previous installation. You can upgrade rules in two ways. Choose the appropriate method depending on how many custom rules are in your Access Manager rule tree.

- If you have many custom rules, migrate your legacy rule tree and then manually add new Teamcenter-supplied rules:

    1. Create a backup of your existing rule tree using the Access Manager **Export** action.

    2. Upgrade your Teamcenter configuration to Teamcenter 13.3. During Teamcenter upgrade, TEM automatically imports your legacy rule tree to Teamcenter 13.3.

    3. Identify changes in the rule tree by comparing the **..**TC_DATA**\\tc_am_rule_tree.default** file in your previous environment to the same file in your Teamcenter 13.3 environment.

    4. Start Access Manager and add rules introduced to Teamcenter since your previous version.

- If you have few or no custom rules, use the standard Teamcenter 13.3 rule tree and then manually add your custom rules.

    1. Create a backup of your existing rule tree using the Access Manager **Export** action.

    2. Identify your custom rules in order to add them after upgrade.

    3. Upgrade your Teamcenter configuration to Teamcenter 13.3. During Teamcenter upgrade, TEM automatically imports your legacy rule tree to Teamcenter 13.3.

    4. Import the standard Teamcenter 13.3 rule tree using the **am_install_tree** utility. Use the **mode=replace_all** argument to overwrite the legacy rule tree with the Teamcenter 13.3 rule tree. The utility automatically creates ACLs and privileges during import.

    5. Manually add your custom rules into the rule tree in the appropriate locations.

The Access Manager supports localization. This includes locale-specific display names of access control list (ACL) objects, privilege names, and accessor type values such as group names and role names. This localization capability is provided using text server XML files. The rule tree import/export functionality supports XML format input files.

The **am_install_tree** utility supports both ASCII text format and XML format rule tree files. However, export in the Access Manager application generates the output file only in XML format. This allows exported ACL name translations to be migrated to other sites.

An XML Access Manager rule tree resembles the following example.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Tc_data_access_config>
  <privileges>
    <priv_name>READ</priv_name>
    <priv_name>WRITE</priv_name>
    <priv_name>COPY</priv_name>
    <priv_name>CHANGE</priv_name>
```

```
        <priv_name>DELETE</priv_name>
    </privileges>

    <named_acls>
      <named_acl>
        <acl_name>Working</acl_name>
        <acl_name language="fr_FR">working_fr</acl_name>
        <acl_name language="de_DE">working_de</acl_name>
        <acl_name language="jp_JP">working_jp</acl_name>
        <ace_entry>
       <accessor_type>group</accessor_type>
       <accessor>dba</accessor>
       <grant>
           <p>READ</p>
           <p>WRITE</p>
           <p>COPY</p>
       </grant>
       <revoke>
           <p>DELETE</p>
           <p>CHANGE</p>
            </revoke>
        </ace_entry>
        <ace_entry>
       <accessor_type>Owning Group</accessor_type>
       <accessor> </accessor>
       <grant>
           <p>READ</p>
           <p>WRITE</p>
           <p>COPY</p>
       </grant>
       <revoke>
           <p>DELETE</p>
           <p>CHANGE</p>
            </revoke>
        </ace_entry>
      </named_acl>
      <named_acl>
        <acl_name>In Project ACL</acl_name>
        <acl_name language="fr_FR">In Project ACL fr</acl_name>
        <acl_name language="de_DE">In Project ACL de</acl_name>
        <acl_name language="jp_JP">In Project ACL jp</acl_name>
        <ace_entry>
          <accessor_type>group</accessor_type>
          <accessor>dba</accessor>
```

Access Manager rule tree example (Continued)

```
        <grant>
           <p>READ</p>
           <p>WRITE</p>
           <p>COPY</p>
        </grant>
            <revoke>
           <p>DELETE</p>
           <p>CHANGE</p>
            </revoke>
         </ace_entry>
         <ace_entry>
           <accessor_type>Owning Group</accessor_type>
           <accessor> </accessor>
           <grant>
           <p>READ</p>
           <p>WRITE</p>
           <p>COPY</p>
        </grant>
            <revoke>
           <p>DELETE</p>
           <p>CHANGE</p>
            </revoke>
         </ace_entry>
      </named_acl>
    </named_acls>
 <rule_tree>
     <tree_node>
        <rule_name>Has Class</rule_name>
        <rule_argument>POM_object</rule_argument>
        <acl_name></acl_name>
        <tree_node>
           <rule_name>Has Bypass</rule_name>
           <rule_argument>true</rule_argument>
           <acl_name>Bypass</acl_name>
        </tree_node>
        <tree_node>
           <rule_name>Has Status</rule_name>
           <rule_argument></rule_argument>
           <acl_name>Vault</acl_name>
        </tree_node>
        <tree_node>
           <rule_name>Has Class</rule_name>
           <rule_argument>POM_application_object</rule_argument>
           <acl_name>Working</acl_name>
           <tree_node>
              <rule_name>Has Class</rule_name>
             <rule_argument>Dataset</rule_argument>
           <acl_name>Dataset ACL</acl_name>
        </tree_node>
         </tree_node>
     </tree_node>
   </rule_tree>
 </Tc_data_access_config>
```

Access Manager rule tree example

2. Reviewing existing access rules

Access Manager rules are considered administration data. Therefore, you can generate an *administration data report* to list the existing access rules on your system using the **generate_admin_data_report** utility. This report captures a snapshot of the configuration at a point in time for archiving or audit reviews. For example, you can review the default administration data for Teamcenter in the Administration Data Documentation report shown below, which is included in the Programming and Customization category of the Teamcenter documentation on Support Center.

The Access Manager rules can be displayed by clicking the **Access Manager** tile.

The Rule Tree lists the rules in the rule tree hierarchy.

A listing of the Access Control Lists (ACLs) are also included.

Rule Tree   **ACLs**   Privileges   Full Tree

Named ACLs (83)

Index	ACL Name	Type	Occurrences
1	Archived Objects	RULETREE	1
2	AuditLog Access	RULETREE	1
3	AuditLog Rule	RULETREE	1
4	Batch Print Dataset	RULETREE	1
5	Batch Print Item	RULETREE	1
6	Batch Print Item Revision	RULETREE	1
7	Bypass	RULETREE	1
8	Classification hierarchy	RULETREE	0
9	Client Session Info	RULETREE	1
10	Closed Edit Context	RULETREE	1
11	Configurator Rule Set ACL	RULETREE	1
12	Digital Sign Dataset	RULETREE	1
13	Digital Sign Item	RULETREE	0
14	Digital Sign Item Revision	RULETREE	0
15	Digital Signature Delete	RULETREE	1
16	Digitally Signed No Write	RULETREE	1
17	EPM Open	RULETREE	0
18	Effectivity Access	RULETREE	1
19	External Data	RULETREE	1
20	GA Working	RULETREE	1
21	General Markup	RULETREE	1
22	IMAN Volume	RULETREE	2
23	Import/Export	RULETREE	1
24	Inactive Edit Context	RULETREE	1
25	Inactive Program	RULETREE	1
26	Inactive Sequence Objects	RULETREE	2
27	Internal Data	RULETREE	1
28	Invalid DS Status No More DS	RULETREE	1

Conditions and accessors are shown, as well.

Administration Data Documentation

Access Manager : Conditions

Report Generation : May 24 2018 07:31 PM IST

Overview   Accessors   **Conditions**   Glossary

Conditions Analysis (87)

Administrative Conditions (1)

Index	Condition	Occurrences	Brief Rule Tree Analysis	Full Rule Tree Analysis
1	Has Bypass	1	Brief Tree	Full Tree

General Conditions (20)

Index	Condition	Occurrences	Brief Rule Tree Analysis	Full Rule Tree Analysis
1	Has Attribute	26	Brief Tree	Full Tree
2	Has Class	86	Brief Tree	Full Tree
3	Has Description	0		
4	Has Form Attribute	0		
5	Has Item ID	0		
6	Has Item Key	0		
7	Has Name	0		
8	Has Node	0		
9	Has Object ACL	4	Brief Tree	Full Tree
10	Has Property	0		
11	Has Status	14	Brief Tree	Full Tree
12	Has Type	16	Brief Tree	Full Tree
13	Has View Type	0		
14	In ICS Hierarchy	0		
15	In Job	1	Brief Tree	Full Tree
16	Inactive Sequence	2	Brief Tree	Full Tree
17	Is Archived	1	Brief Tree	Full Tree
18	Is Local	0		
19	Site Geography	0		
20	User TTC Expired	0		

Ownership/Accessor based Conditions (6)

Index	Condition	Occurrences	Brief Rule Tree Analysis	Full Rule Tree Analysis
1	Is GA	1	Brief Tree	Full Tree
2	Is SA	0		

Overview   **Accessors**   Conditions   Glossary

Accessor Type Analysis (42)

General Accessor Types (13)

Index	Accessor Type	Occurrences	Brief Rule Tree Analysis	Full Rule Tree Analysis
1	Group	4	Brief Tree	Full Tree
2	Group Administrator	0		
3	Groups With Security	12	Brief Tree	Full Tree
4	Owning Group	42	Brief Tree	Full Tree
5	Owning User	59	Brief Tree	Full Tree
6	Remote Site	1	Brief Tree	Full Tree
7	Role	18	Brief Tree	Full Tree
8	Role in Group	0		
9	Role in Owning Group	0		
10	Site	0		
11	System Administrator	47	Brief Tree	Full Tree
12	User	4	Brief Tree	Full Tree
13	World	117	Brief Tree	Full Tree

Workflow Accessor Types (7)

Index	Accessor Type	Occurrences	Brief Rule Tree Analysis	Full Rule Tree Analysis
1	Approver	0		
2	Approver(Group)	0		
3	Approver(RIG)	0		
4	Approver(Role)	0		
5	Responsible Party	0		
6	Task Owner	0		
7	Task Owning Group	0		

Project Accessor Types (6)

Index	Accessor Type	Occurrences	Brief Rule Tree Analysis	Full Rule Tree Analysis
1	Current Project Team	0		
2	Current Project Teams	0		
3	Project Team	0		
4	Project Teams	1	Brief Tree	Full Tree
5	Role in Project	0		
6	Role in Projects of Object	0		

You can generate an Administration Data Report specific to your site using the **generate_admin_data_report** utility.

3. Creating and managing rules

Creating and managing Access Manager rules

The Access Manager (AM) rule tree determines privileges on objects in the database. You must have system administrator privileges to modify the AM rule tree.

Understanding how rules work

How rules are defined

Rules are defined by a combination of a condition, a value for that condition, and an access control list (ACL) that grants privileges to accessors.

- The condition and value identify the set of objects to which the rule applies.

- The ACL defines the privileges that are granted to users (accessors) specified in the ACL.

IF *condition = value* is **TRUE, THEN** apply ACL to object.

Example ACL

Accessor	User	Read	Write	Delete	Change	Promote	Demote	Copy
World		✔	✘	✘	✘	✘	✘	✔

Rule syntax

The following syntax applies to rules:

> Condition {Value} –> ACL

The parts of the rule can be thought of as an **IF** clause and a **THEN** clause.

- The condition and value supply the **IF** part of the rule and examine the object with Boolean logic.

- The access control list (ACL) supplies the **THEN** part of the rule by describing the access permission.

For example:

Has Type {UGMASTER} –> UG Model

In this example, **Has Type** is the condition, **UGMASTER** is the value, and **UG Model** is the name of the ACL.

Rule evaluation assumptions

When a user attempts to access data, the rule tree is evaluated to determine the privileges to be granted or denied. The following assumptions apply to the evaluation:

- Rules higher in the rule tree are more global in nature and apply to all object types.

- Lower-level rules refine access to more specific objects such as **UGMASTER** datasets. For example:

    Has Class(POM_application_object)

        Has Class(Dataset)

            Has Type(UGMASTER)

- **Precedence** determines the privileges granted. Rule precedence is from top to bottom in the tree, with the highest rule having greatest precedence and the lowest rule having least precedence.

- **Accessor precedence** in the ACL and rule precedence within the tree are both considered when granting access privileges. Accessors have a predefined precedence in the system.

> Note:
>
> The way Access Manager evaluates Master forms does not follow the normal rules. Master forms inherit access privileges from the parent item or item revision, so if you change access privileges to an item or item revision, you affect the privileges on the Master form. You can use the **TC_MASTERFORM_DELEGATE** environment variable to change the default behavior.

Evaluating the rule tree for the effective ACL

The rule tree evaluation results in an *effective ACL*. The effective ACL represents the cumulative compilation of all the named ACLs that apply to the object the user is trying to access.

The rule tree is evaluated as follows:

- Trim rules that do not apply to the object because their conditions are false.

  > Note:
  >
  > The rules are not removed from the tree, but they are ignored during evaluation.

- Evaluate rules in order of precedence, from top to bottom.

- Evaluate the subbranch of a rule before evaluating the parent rule.

- Evaluate subbranch rules in order of precedence, from top to bottom, in the event that there are multiple subbranch rules.

The *effective ACL* is determined by compiling the ACLs in the order that the tree is traversed.

Example rule tree evaluation by order of precedence

This example rule tree shows the order of precedence in the left column, assuming all conditions are met.

- The first two rows are the first two rules evaluated because they are highest in the tree and have no subbranch.

- The third row only gets evaluated after all its subbranches are evaluated.

```
1      Condition {Value} -> Named ACL
2      Condition {Value} -> Named ACL
15  -  Condition {Value} -> Named ACL
9      -  Condition {Value} -> Named ACL
3            Condition {Value} -> Named ACL
4            Condition {Value} -> Named ACL
7          -  Condition {Value} -> Named ACL
5               Condition {Value} -> Named ACL
6               Condition {Value} -> Named ACL
8            Condition {Value} -> Named ACL
14     -  Condition {Value} -> Named ACL
10           Condition {Value} -> Named ACL
13         -  Condition {Value} -> Named ACL
11               Condition {Value} -> Named ACL
12               Condition {Value} -> Named ACL
```

Example of compiling an effective ACL

When the user attempts to access a **UGMASTER** dataset, the **rule tree** is trimmed to reflect only those rules that apply to the object.

Has Class(POM_object)

   Has Class(POM_application_object) –> Working

Has Class(Dataset)

Has Type(UGMASTER) –> UGMASTER

Based on the trimmed rule tree, the effective ACL is compiled by evaluating the tree (from bottom to top) as follows:

1. Find the topmost leaf node in the tree, in this case, **Has Type(UGMASTER) –> UGMASTER**. Add the **UGMASTER** ACL to the effective ACL.

2. Find the next node, **Has Class(Dataset)**. This node has no associated ACL, so it does not contribute to the effective ACL.

3. Find the next node, **Has Class(POM_application_object) –> Working**. Add the **Working** ACL to the effective ACL.

4. Find the next node, **Has Class(POM_object)**. This node has no associated ACL, so it does not contribute to the effective ACL.

The rule tree evaluation results in the following effective ACL.

Accessor	User	Read	Write	Delete	Change	Promote	Demote	Copy	ACL
Role in Owning Group	Designer		✔					✔	UGMASTER
World			✘		✘			✘	UGMASTER
Owning User			✔	✔	✔				Working
Group Administrator				✔	✔				Working
Owning Group			✔						Working
System Administrator				✔	✔				Working
World		✔	✘	✘	✘	✘	✘	✔	Working

The effective ACL is evaluated when a user attempts to access a **UGMASTER** dataset. The lines that do not apply to the user are ignored. For example, if you are a designer in the owning group of the **UGMASTER** dataset, but you are not the owning user, system administrator, or group administrator, the following entries in the ACL are applied when you try to access a **UGMASTER** dataset.

Accessor	User	Read	Write	Delete	Change	Promote	Demote	Copy
Role in Owning Group	Designer		✔					✔
World			✘		✘			✘
World		✔		✘		✘	✘	

After the effective ACL is trimmed to include only the entries that apply to the user attempting to access the dataset, the privileges in the remaining ACL entries are evaluated. This is done by working down each privilege column until you encounter a granted ✔ or denied ✘ symbol.

In this example, the privilege evaluation grants the accessor read, write, and copy privileges and denies the accessor delete, change, promote, and demote privileges.

Simple rule tree evaluation example

This simplified view of the default rule tree is used in the following example:

```
Has Class(POM_object)
  Has Bypass(true) –> Bypass
  Has Status( ) –> Vault
  Has Class(POM_application_object) –> Import/Export
```

A user, Jim Smith, attempts to open the **MyDataset** text dataset with released status. To perform this action, Jim Smith needs read privileges on the dataset.

The following ACLs are considered when the sample rule tree is evaluated:

1. The **Has Bypass(true) –> Bypass** rule is evaluated. This high-level rule grants system administration privileges to users.

    Result: Jim does not have bypass set, nor is he a system administrator; therefore, this rule condition is false and the **Bypass** ACL is not applied. The evaluation moves down the tree to the next branch.

2. The **Has Status() –> Vault** rule is evaluated. This rule evaluates whether the object has an attached status type. If yes, the **Vault** ACL is applied.

    Result: The **MyDataset** dataset is in released status; therefore, the rule condition is true and the **Vault** ACL is applied.

Vault ACL

The **Vault** ACL grants all users read and copy privileges and denies all users write, delete, change, promote, and demote privileges. The **World** accessor represents all users.

Accessor	User	Read	Write	Delete	Change	Promote	Demote	Copy	CICO
World		✔	✘	✘	✘	✘	✘	✔	✘

3. The **Has Class(POM_application_object_) –> Import/Export** rule is evaluated. This rule evaluates whether the object is of the **POM_application_object** class. If yes, the **Import/Export** ACL is applied to the object.

   Result: All workspace objects, including datasets, are subclasses of the **POM_application_object** class; therefore, the rule condition is true and the **Import/Export** ACL is applied.

   Import/Export ACL

   The **Import/Export** ACL grants all users (world) export, import and transfer in privileges and denies all users transfer out privileges. In addition, this ACL grants remote site users import privileges and denies remote site users transfer in privileges. The **Import/Export** ACL neither explicitly grants or denies read privileges.

Accessor	User	Read	Export	Import	Transfer out	Transfer in
World			✔	✔	✘	✔
Remote Site				✔		✘

Complex rule tree example

This view of the default rule tree is used in the example that follows:

> Has Class(POM_object)
>
> > Has Bypass(true) –> Bypass
> >
> > In Job(true)
> >
> > Has Status( ) –> Vault

Has Object ACL(true)

Has Class(POM_application_object) –> Working

    Has Class(Item) –> Items

    Has Class(Item Revision) –> Item Revs

    Has Class(Dataset)

        Has Type(UGMASTER) –> UGMASTER
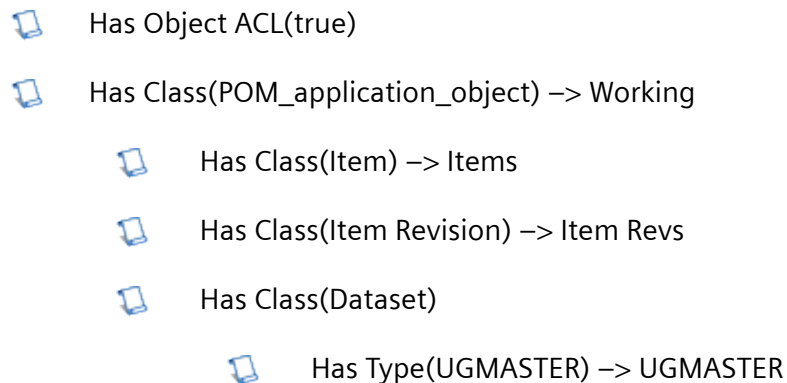
A user, Jim Smith (**jsmith**), a designer in the engineering group, attempts to modify the **MyPart UGMASTER** dataset with working status. To perform this action, Jim Smith needs write privileges on the dataset.

The following ACLs are considered when the sample rule tree is evaluated:

1.   The **Has Bypass(true) –> Bypass** rule is evaluated. This high-level rule grants system administration privileges to users.

    Result: Jim does not have bypass set, nor is he a system administrator; therefore, this rule condition is false and the **Bypass** ACL is not applied. The evaluation moves down the tree to the next branch.

2.   The **In Job(true)** rule is evaluated. This rule evaluates whether the object is in a workflow.

    Result: No ACL is defined, therefore, the condition being true has no effect. The evaluation moves down the tree to the next branch.

3.   The **Has Status() –> Vault** rule is evaluated. This rule evaluates whether the object has an attached status type. If yes, the **Vault** ACL is applied.

    Result: The **MyPart** dataset is in working status; therefore, the rule condition is false and the **Vault** ACL is not applied.

4.   The **Has Object ACL(true)** rule is evaluated. This rule evaluates whether an ACL exists for the object.

    Result: No object ACL is defined by a user; therefore, the condition is false and has no effect. The evaluation moves down the tree to the next branch.

5.   The **Has Class(Item) –> Items** rule is evaluated. This rule evaluates whether the object is of class item. If yes, the **Items** ACL is applied.

Result: The **MyPart** is of class dataset not item; therefore, the rule condition is false and the **Items** ACL is not applied.

6. The **Has Class(Item Revision) –> Item Revs** rule is evaluated. This rule evaluates whether the object is of class item revision. If yes, the **Items** ACL is applied.

   Result: The **MyPart** dataset is of class dataset not item revision; therefore, the rule condition is false and the **Item Revs** ACL is not applied.

7. The **Has Type(UGMASTER) –> UGMASTER** rule is evaluated. This rule evaluates whether the object is of class UGMASTER. If yes, the Items ACL is applied.

   Result: The **MyPart** dataset is of class **UGMASTER**; therefore, the rule condition is true and the **UGMASTER** ACL is applied.

   UGMASTER ACL

   The **UGMASTER** ACL explicitly grants write access to users who fill the **Designer** role in the owning group and explicitly denies write access to all other users in the owning group.

Accessor	User	Read	Write	Delete	Change	Promote	Demote	Copy
Role in Owning Group	Designer		✔					
Owning Group			✘					

8. The **Has Class(Dataset)** rule is evaluated. This rule evaluates whether the object is of class dataset.

   Result: The **MyPart** dataset is of class dataset; therefore, the rule condition is true. No ACL is defined, therefore the condition being true has no effect.

9. The **Has Class(POM_application_object) –> Working** rule is evaluated. This rule evaluates whether the object is of the **POM_application_object** class. If yes, the **Working** ACL is applied to the object.

   Result: All workspace objects, including datasets, are subclasses of the **POM_application_object** class; therefore, the rule condition is true and the **Working** ACL is applied.

   Working ACL

   The **Working** ACL explicitly grants write, delete, and change privileges to owning users and write privileges to the owning group. It also grants delete and change privileges to the group

administrator and the system administrator. All other users are granted read and copy privileges and explicitly denied write, delete, change, promote, and demote privileges.

Accessor	User	Read	Write	Delete	Change	Promote	Demote	Copy
Owning User			✔	✔	✔			
Group Administrator				✔	✔			
Owning Group			✔					
System Administrator				✔	✔			
World		✔	✖	✖	✖	✖	✖	✔

Result: After all the rules are evaluated, the following is the result. Note that the **Working** ACL grants the owning group write permission, but the **UGMASTER** ACL already removed that privilege. The figure also shows the applied named ACL.

Accessor	User	Read	Write	Delete	Change	Promote	Demote	Copy	Named ACL
World									Import /Export
Remote Site									Import/ Export
Role in Owning Group	Designer		✔						UGMASTER
Owning Group			✖						UGMASTER
Owning User			✔	✔	✔				Working
User	tsproxy (tsproxy)		✔	✔					Working
Group Administrator				✔	✔				Working
Owning Group			✔						Working
System Administrator				✔	✔				Working
World		✔	✖	✖	✖	✖	✖	✔	Working

Understanding the rule creation process

1. **Add a rule to the tree**.

2. **Create and save the access control list (ACL)**.

3. **Attach the new ACL to the rule by modifying the rule**.

> Tip:
>
> You must always save the rule or ACL after making modifications.

Access Manager conditions

Access conditions by group

The following table lists the access conditions by category. Click a condition to learn more about it.

Condition	Description
Administrative	
Has Application	Provides additional security to administration applications, for example, Organization, Access Manager, and Authorization.
Has Bypass	Specifies whether the user has bypass privileges set. Bypass privilege supersedes other privileges. This privilege allows administrators to make changes that could potentially cause unintended loss of data and have serious repercussions that are normally guarded against by access rules.
Has Metadata Class	Provides additional security to property conditions and other metadata.
General	
Has Attribute	Specifies an attribute and value associated with a particular class.
Has Class	Specifies an object class. The object is evaluated to determine if it is of the specified class.
Has Classification	Validates the custom classification attribute value of the object against the value specified for the condition.
Has Description	Specifies a description for the object. The object is evaluated to determine whether the description matches this value.
Has Digital Signature	Specifies whether a business object has a digital signature of the specified status.
Has Form Attribute	Enables access control of items and item revisions by setting conditions on attributes of the **Masterform** class.
Has Item ID	Specifies an item ID against which the item is evaluated.
Has Item Key	Specifies a multifield key identifier against which the item is evaluated.
Has Name	Specifies a name against which the object is evaluated.

Condition	Description
Has Object ACL	Specifies that an ACL is associated with an object. This condition does not expect an ACL attached to a rule. It is a placeholder that indicates the point at which process ACLs and object ACLs are applied in the rule tree hierarchy.
Has Property	Specifies the value of a compound property against which an object is evaluated.
Has Status	Specifies the status type against which the object is evaluated.
Has Type	Specifies the object type against which the object is evaluated.
Inactive Sequence	Specifies that previous sequences are historical and cannot be worked on independently. The latest sequence is always the working sequence for the revision.

> **Note:**
>
> This condition is used in conjunction with the **Inactive Sequence Objects** ACL.

Condition	Description
In Job	Specifies whether the target object is in a workflow job (process). This condition does not expect an ACL attached to a rule. It is a placeholder that indicates the point at which workflow ACLs are applied in the rule tree hierarchy.

> **Note:**
>
> No subbranches can be added below the **In Job** branch in the Access Manager rule tree.

Condition	Description
Is Archived	Specifies that the object's archive status is evaluated.
Is Local	Specifies whether the object's residence in the local database is evaluated. This condition is used when Multi-Site Collaboration is implemented.
Is Sponsored Mode	Checks whether the Teamcenter session is in sponsored mode. It enables end users to configure rules to enforce data access control when the Teamcenter session is launched in sponsored mode.
Site Geography	Checks whether the given geography matches the geography of the site being evaluated.
User Has Digital Signature	Specifies whether a business object has a digital signature of the specified status in the context of the logged-on user.
Ownership/Accessor based	
Current Group Is	Checks the current logged-on group that is set in the session. It enables end users to configure access rules for the **Sponsor** group.
Is Current Group External	Evaluates whether the security of the current logged in group is external.
Is GA	Specifies whether the user's status as a group administrator in the current group is evaluated.
Is Group External	Evaluates whether the object under consideration is Group object and has external security.
Is Group Member External	Evaluates whether the object under consideration is GroupMember and belongs to a group that has external security.
Is Group Same As Current Group	Evaluates whether the object under consideration is Group and is the same as the current logged in group that has external security.

Condition	Description
Is Member Group Same As Current Group	Evaluates whether the group member object belongs to the same group as the current logged on group.
Is SA	Specifies whether the user's system administration group membership is evaluated.
Owning Group	Evaluates whether the object is owned by the group under which the user is logged on to Teamcenter.
Owning Group Has Security	Evaluates whether the owning group of the object has a security string. This condition is true only if the security value of the owning group is equal to the value of this condition.
Owning Site	Evaluates whether the object is owned by the specified site. This condition is used when Multi-Site Collaboration is implemented.
Owning User	Evaluates whether the object is owned by the specified user.
Is User External	Evaluates whether the user object is from a group whose security is external.
Is User In Current Group	Evaluates whether the user object under evaluation has current group membership.
Incremental Change	
In IC Context	Enables structure edits (occurrence edits, occurrence notes, transform edits, and attachment edits) to be controlled by the Structure Manager, Manufacturing Process Planner, Multi-Structure Manager, or Part Planner application.
Project	
In Current Project	Specifies the project ID against which the object is evaluated.
	Note: This rule is not delivered with the default installation of Teamcenter. It must be added manually.
In Project	Specifies a project to which the object must be assigned.
Is Project Member	Specifies whether the user's membership in the project is evaluated. This condition is only true when the user is a current member of the project.
Has Project Of Category	Checks whether the workspace object being evaluated has any project assigned of the given category.
Program	
In Current Program	Specifies access based on whether the program to which the data is assigned is the current program under which the user is logged on to Teamcenter.
In Inactive Program	Controls access to data based on whether the status of the owning program is **inactive**.
In Invisible Program	Controls access to data based on whether the status of the owning program is **invisible**.
Is Owned By Program	Controls access to data based on whether data is owned by the program specified as a value for the **Is Owned By Program** condition.
Is Program Member	Specifies whether the user's membership in the program is evaluated. This condition is only true when the user is a member of the owning program or a shared program.
General Authorized data access (ADA) licenses	

Condition	Description
ADA License Has Citizenship	Checks whether the ADA license being evaluated has the given citizenship.
Citizenship On Any ADA Lic	Checks whether the citizenship of the user being evaluated matches any of the citizenships applied to the ADA licenses attached to the workspace objects.
Has ADA License Of Category	Checks whether the workspace object being evaluated has any ADA license of the given category.
Has Named ADA License	Checks whether a specific ADA license is attached to the workspace objects being evaluated.
User In Attach ADA Lic of Ctgry	Checks whether the user being evaluated is listed in the ADA license attached to the workspace objects. The given category must match that on the ADA license.
User In Attached License	Checks whether the user being evaluated is listed on any or all of the ADA licenses attached to the workspace objects.
User In License	Verifies that the user being evaluated is listed in the ADA license.
User In Named License	Checks whether the user being evaluated is listed on an ADA license of the specified name. It does not check if the license is attached to the workspace objects being evaluated.
User-ADA Lic Has Citizenship	Checks whether the user's citizenship matches the passed-in value and then sees if the user's citizenship is on any of the ADA licenses attached to the workspace object being evaluated.
International Traffic in Arms Regulations (ITAR)	
Citizenship On Any ITAR Lic	Checks whether a citizenship of the user being evaluated matches any of the citizenships applied to the ITAR licenses attached to the workspace objects.
Group Nationality	Checks whether the given nationality matches the group nationality.
Has Government Classification	Compares the classification level in the condition argument with the object classification level. If the object is not classified, or if the object classification level is less than that of the given classification in the argument, this condition returns **True**.
Has ITAR License Of Category	Checks whether the workspace object being evaluated has any ITAR license of the given category.
Has Named ITAR License	Checks whether a specific ITAR license is attached to the workspace objects being evaluated.
Has No Government Classification	Checks if there is no government classification value on the workspace object.
ITAR License Has Citizenship	Checks whether the ITAR license being evaluated has the given citizenship.
Site Geography	Checks whether the given geography matches the geography of the site being evaluated.
User Citizenship	Checks whether the given citizenship matches the citizenships of the user being evaluated.
User Citizenship Or Nationality	Checks whether the given citizenship matches the citizenship or nationality of the user being evaluated.
User Declared Geography	Checks whether the given geography matches the geography the user declared when logging on to the system.

Condition	Description
	Note: For more information about **User Declared Geography**, see Configure geography access.
User Geography	Checks whether the given geography matches the geography of the user being evaluated.
User Has Government Clearance	Checks whether the government classification level of the user being evaluated is equal to, greater than, or less than the value specified in the condition.
User In Attach ITAR Lic of Ctgry	Checks whether the user being evaluated is listed in the ITAR licenses attached to the workspace objects. The given category must match that on the ITAR license.
User In Attached ITAR License	Checks whether the user being evaluated is listed on any or all of the ITAR licenses attached to the workspace objects.
User In Named ITAR License	Checks whether the user being evaluated is listed on an ITAR license of the specified name. It does not check if the license is attached to the workspace objects being evaluated.
User Is ITAR Licensed	Checks whether the user currently logged on is cited in a valid (not expired) ITAR license attached to the workspace object either directly or by membership in a cited organization (group).
User Nationality	Checks whether the given nationality matches the nationality of the user being evaluated.
User TTC Expired	Checks whether the current date is later than the technology transfer certification (TTC) date on the **User** object.
User-ITAR Lic Has Citizenship	Checks whether the user's citizenship matches the passed-in value and then sees if the user's citizenship is on any of the ITAR licenses attached to the workspace object being evaluated.
Intellectual property (IP) license	
Citizenship On Any IP Lic	Checks whether the citizenship of the user being evaluated matches any of the citizenships applied to the IP licenses attached to the workspace objects.
Has IP Classification	Checks whether the IP classification of the workspace object being evaluated is equal to, greater than, or less than the value specified in the condition.
Has IP License Of Category	Checks whether the workspace object being evaluated has any IP license of the given category.
Has Named IP License	Checks whether a specific IP license is attached to the workspace objects being evaluated.
Has No IP Classification	Checks whether the workspace object does not have a value specified in the IP classification attribute.
IP License Has Citizenship	Checks whether the IP license being evaluated has the given citizenship.
User Has IP Clearance	Checks whether the IP clearance level of the user being evaluated is equal to, greater than, or less than the value specified in the condition.
User In Attach IP Lic of Ctgry	Checks whether the user being evaluated is listed in the IP license attached to the workspace objects. The given category must match that on the IP license.
User In Attached IP License	Checks whether the user being evaluated is listed on any or all of the IP licenses attached to the workspace objects.

Condition	Description
User In Named IP License	Checks whether the user being evaluated is listed on an IP license of the specified name. It does not check if the license is attached to the workspace objects being evaluated.
User Is IP Licensed	Checks whether the user being evaluated is listed on an IP license attached to the workspace object.
User-IP Lic Has Citizenship	Checks whether the user's citizenship matches the passed-in value and then sees if the user's citizenship is on any of the IP licenses attached to the workspace object being evaluated.
Exclude licenses	
Citizenship On Any Exclude Lic	Checks whether the citizenship of the user being evaluated matches any of the citizenships applied to the exclude licenses attached to the workspace objects.
Exclude License Has Citizenship	Checks whether the exclude license being evaluated has the given citizenship.
Has Exclude License Of Category	Checks whether the workspace object being evaluated has any exclude license of the given category.
Has Named Exclude License	Checks whether a specific exclude license is attached to the workspace objects being evaluated.
User In Attach Excl Lic of Ctgry	Checks whether the user being evaluated is listed in the exclude license attached to the workspace objects. The given category must match that on the exclude license.
User In Attached Exclude License	Checks whether the user being evaluated is listed on any or all of the exclude licenses attached to the workspace objects.
User In Named Exclude License	Checks whether the user being evaluated is listed on an exclude license of the specified name. It does not check if the license is attached to the workspace objects being evaluated.
User Is Excluded	Checks whether the user being evaluated is listed on an exclude license attached to the workspace object.
User-Exclude Lic Has Citizenship	Checks whether the user's citizenship matches the passed-in value and then sees if the user's citizenship is on any of the exclude licenses attached to the workspace object being evaluated.

ADA License Has Citizenship

CATEGORY

License by Category

DESCRIPTION

Checks whether the ADA license being evaluated has the given citizenship.

> Note:
>
> Citizenships are a two-letter country code from ISO 3166 (for example, Germany's country code is **DE**). A user can have multiple citizenships.

CONDITION EVALUATION

true	If any of the citizenships of the ADA license being evaluated match the specified citizenship, the condition evaluates to **true**.
false	If none of the citizenships of the ADA license being evaluated match the specified citizenship, the condition evaluates to **false**.

INPUT ARGUMENTS

(Custom License:_citizenship_**)**	Two-character ISO 3166 codes identifying a country.
	This condition accepts negation using a minus (**–**) prefix. For example, **–IR** means that the user cannot have an IR citizenship.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

• Any workspace object.

Citizenship On Any ADA Lic

CATEGORY

International Traffic in Arms Regulations (ITAR)

DESCRIPTION

Checks whether any or all of the citizenships of the user being evaluated matches any of the citizenships on the ADA licenses attached to the workspace objects.

> Note:
>
> Citizenships are a two-letter country code from ISO 3166 (for example, Germany's country code is **DE**). A user can have multiple citizenships.

CONDITION EVALUATION

true
- If set to **Any**, the condition evaluates to **true** if any citizenship of the user being evaluated matches the user citizenships applied to any nonexpired ADA licenses attached to the workspace object being evaluated.

- If set to **All**, the condition evaluates to **true** if all of the citizenships listed for the user being evaluated are found on any nonexpired ADA licenses. Each of the user's citizenships must be on at least one of the nonexpired ADA licenses but does not have to be on each nonexpired ADA license.

false
- If set to **Any**, the condition evaluates to **false** if none of the citizenships of the user being evaluated match the user citizenships applied to any nonexpired ADA license attached to workspace object being evaluated.

- If set to **All**, the condition evaluates to **false** if at least one of the citizenships listed for the user being evaluated is not found on any nonexpired ADA licenses.

INPUT ARGUMENTS

- **Any**

- **All**

- (Custom License:{**Any|All**})

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Any workspace object.

RELATED RULE CONDITIONS

- **Citizenship On Any Exclude Lic**

- **Citizenship On Any IP Lic**

- **Citizenship On Any ITAR Lic**

Citizenship On Any Exclude Lic

CATEGORY

International Traffic in Arms Regulations (ITAR)

DESCRIPTION

Checks whether the citizenship of the user being evaluated matches any of the citizenships applied to the exclude licenses attached to the workspace objects.

> Note:
>
> Citizenships are a two-letter country code from ISO 3166 (for example, Germany's country code is **DE**). A user can have multiple citizenships.

CONDITION EVALUATION

true
- If set to **Any**, the condition evaluates to **true** if any citizenship of the user being evaluated matches the user citizenships applied to any of the nonexpired exclude licenses attached to the workspace object being evaluated.

- If set to **All**, the condition evaluates to **true** if all of the citizenships of the user being evaluated match any of the user citizenships applied to the nonexpired exclude licenses attached to the workspace object being evaluated. Each of the user citizenships must be on at least one of the nonexpired exclude licenses but does not have to be on each nonexpired exclude license.

false
- If set to **Any**, the condition evaluates to **false** if none of the citizenships of the user being evaluated matches the user citizenships applied to any of the nonexpired exclude licenses attached to workspace object being evaluated.

- If set to **All**, the condition evaluates to **false** if at least one of the citizenships of the user being evaluated is not found on any of the nonexpired exclude licenses attached to workspace object being evaluated.

INPUT ARGUMENTS

Any or **All**

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Any workspace object.

RELATED RULE CONDITIONS

- **Citizenship On Any ADA Lic**

- **Citizenship On Any IP Lic**

- **Citizenship On Any ITAR Lic**

Citizenship On Any IP Lic

CATEGORY

Intellectual Property (IP)

DESCRIPTION

Checks whether any or all of the citizenships of the user being evaluated matches any of the citizenships on the IP licenses attached to the workspace objects.

> Note:
>
> Citizenships are a two-letter country code from ISO 3166 (for example, Germany's country code is **DE**). A user can have multiple citizenships.

CONDITION EVALUATION

true
- If set to **Any**, the condition evaluates to **true** if any citizenship of the user being evaluated matches the user citizenships applied to any of the nonexpired IP licenses attached to the workspace objects.

- If set to **All**, the condition evaluates to **true** if all of the citizenships of the user being evaluated matches the user citizenships of any nonexpired IP licenses attached to the workspace objects. Each of the user citizenships must be on at least one of the nonexpired IP licenses but does not have to be on each nonexpired IP license.

false
- If set to **Any**, the condition evaluates to **false** if none of the citizenships of the user being evaluated match the user citizenships applied to any of the nonexpired IP license attached to the workspace object being evaluated.

- If set to **All**, the condition evaluates to **false** if at least one of the citizenships of the user being evaluated is not found on any nonexpired IP licenses.

INPUT ARGUMENTS

Any or **All**

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Any workspace object.

RELATED RULE CONDITIONS

- **Citizenship On Any ADA Lic**

- **Citizenship On Any Exclude Lic**

- **Citizenship On Any ITAR Lic**

Citizenship On Any ITAR Lic

CATEGORY

International Traffic in Arms Regulations (ITAR)

DESCRIPTION

Checks whether a citizenship of the user being evaluated matches the any of the citizenships applied to the ITAR licenses attached to the workspace objects.

> Note:
>
> Citizenships are a two-letter country code from ISO 3166 (for example, Germany's country code is **DE**). A user can have multiple citizenships.

CONDITION EVALUATION

true
- If set to **Any**, the condition evaluates to **true** if any citizenship of the user being evaluated matches the user citizenships applied to any of the nonexpired ITAR licenses attached to the workspace objects.

- If set to **All**, the condition evaluates to **true** if all of the citizenships of the user being evaluated are found on any of the nonexpired ITAR licenses attached to the workspace objects. Each of the user citizenships must be on at least one of the nonexpired ITAR licenses but does not have to be on each nonexpired ITAR license.

- If none of the nonexpired ITAR licenses attached to the workspace objects have user citizenships applied, the condition evaluates to **true**.

false
- If set to **Any**, the condition evaluates to **false** if none of the citizenships of the user being evaluated matches the user citizenships applied of any nonexpired ITAR license attached to workspace object being evaluated.

- If set to **All**, the condition evaluates to **false** if at least one of the citizenships of the user being evaluated is not found on any nonexpired ITAR licenses.

INPUT ARGUMENTS

Any or **All**

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Any workspace object.

RELATED RULE CONDITIONS

- **Citizenship On Any ADA Lic**

- **Citizenship On Any Exclude Lic**

- **Citizenship On Any IP Lic**

Current Group Is

CATEGORY

General

DESCRIPTION

Checks the current logged-on group that is set in the session. It enables end users to configure access rules for the **Sponsor** group.

> Note:
>
> This condition applies to the current logged-on user only. This does not apply to a given user and group that are different from the logged-on user group.

INPUT ARGUMENTS

true or **false**

RELATED RULE CONDITIONS

- **Is Sponsored Mode**

Exclude License Has Citizenship

CATEGORY

License by Category

DESCRIPTION

Checks whether the IP license being evaluated has the given citizenship.

> Note:
>
> Citizenships are a two-letter country code from ISO 3166 (for example, Germany's country code is **DE**). A user can have multiple citizenships.

CONDITION EVALUATION

true   If any of the citizenships of the user being evaluated match the specified citizenship, the condition evaluates to **true**.

false   If none of the citizenships of the user being evaluated match the specified citizenship, the condition evaluates to **false**.

INPUT ARGUMENTS

citizenship   Two-character ISO 3166 codes identifying a country.

This condition accepts negation using a minus (**–**) prefix. For example, **–IR** means that the user cannot have an IR citizenship.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

• Any workspace object.

Group Nationality

CATEGORY

International Traffic in Arms Regulations (ITAR)

DESCRIPTION

Checks whether the given nationality matches the nationality of the group being evaluated.

CONDITION EVALUATION

true	This condition evaluates to **true** if the given nationality matches the nationality of the group being evaluated.
false	In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

nationality	Two-character ISO 3166 country codes identifying the nationality of the group or organization.
	This condition accepts multiple nationality arguments, which must be comma separated as shown in the Example section.
	Also, this condition accepts negation using a minus (**-**) prefix. For example, **-US** indicates any group outside the U.S.
	When negating multiple nationality arguments, you must enclose the nationality arguments in parentheses. For example, **-(US,FR)** indicates the group is not authorized in the U.S. and France.

BUSINESS OBJECT SCOPE

This condition can be used to control access to classified data.

EXAMPLE

- To specify the nationality of a group in the U.S., enter **US**.

- To specify the nationality of a group *outside* the U.S., enter **-US**.

- To specify the nationality of a group in multiple countries (such as U.S., France, Japan, and India), enter **US,FR,JP,IN**.

- To specify the nationality of a group *outside* multiple countries (such as U.S., France, Japan, and India), enter **-(US,FR,JP,IN)**.

RELATED RULE CONDITIONS

- **User Nationality**

Has ADA License Of Category

CATEGORY

License by Category

DESCRIPTION

Checks if any type of Authorized Data Access (ADA) license with the specified category is attached to the workspace object being evaluated.

CONDITION EVALUATION

true	If there is any type of ADA license with the specified category attached to the workspace object, this condition evaluates to **true**.
false	If there is no ADA license with the specified category or if the license exists but is not attached to the workspace object, the condition evaluates to **false**.

INPUT ARGUMENTS

(Custom License:license_category**)**   A string identifying the category of the license.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

• Any workspace object.

EXAMPLE

For an example, see Security Administration.

RELATED RULE CONDITIONS

• **Has Exclude License Of Category**

• **Has IP License Of Category**

• **Has ITAR License Of Category**

Has Application

CATEGORY

Administrative

DESCRIPTION

Provides additional security to administration applications, for example, Organization, Access Manager, and Authorization. Therefore, if any user gets unauthorized access to these applications, access is denied to that user, as the **World** accessor is explicitly granted **Read** privileges and explicitly denied **Write**, **Delete**, **Check-In/Check-Out**, and **Create** access control list (ACL) privileges.
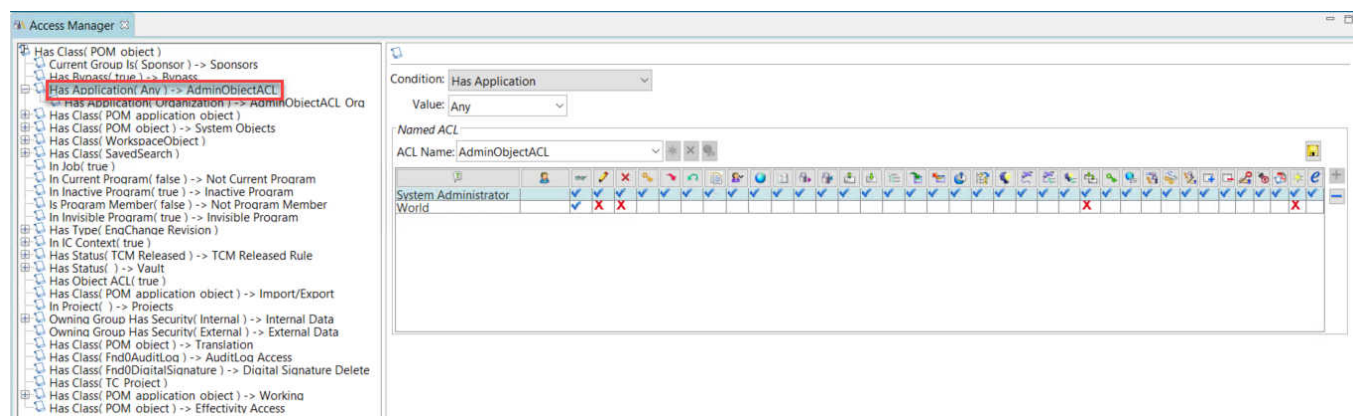
For example, a non-dba user can access the Authorization application, but is only granted read access.
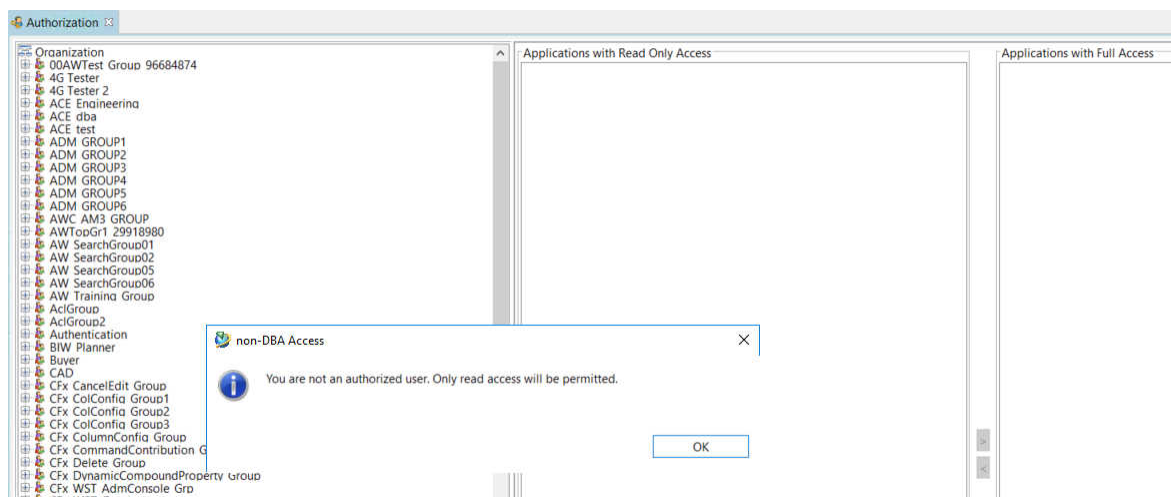
INPUT ARGUMENTS

Any
AccessManager

ADALicense
AuditManager
Authorization
ClassificationAdministration
LibraryManagementAdministration
Organization
Subscription

> Note:
>
> If you select **Any** as an input argument, it includes all applications registered with Access Manager.

EXAMPLES

- **Example 1**

  As a customer, you want to restrict access to the Organization application to only allow users with **DBA** privileges and your research and development group users to access your data. To achieve this use case, create rules in the Access Manager rule tree to grant **Read**, **Write**, **Delete**, **Check-In/Check-Out**, and **Create** access for any non-dba user.

  1. Using Organization, create a group, for example, **AclGroup**, and create a role, for example, **AclRole**, and add it to the **AclGroup**. Then, add your research and development group users, for example, **RD_engineer**, to the **AclRole**.
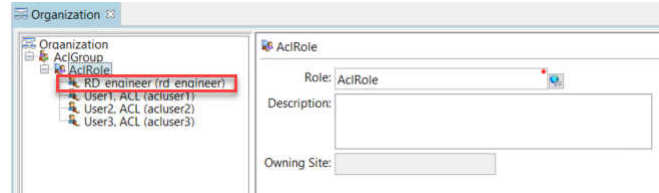
  2. Using Authorization, set your new role, **AclRole**, to have full access to the Organization application.
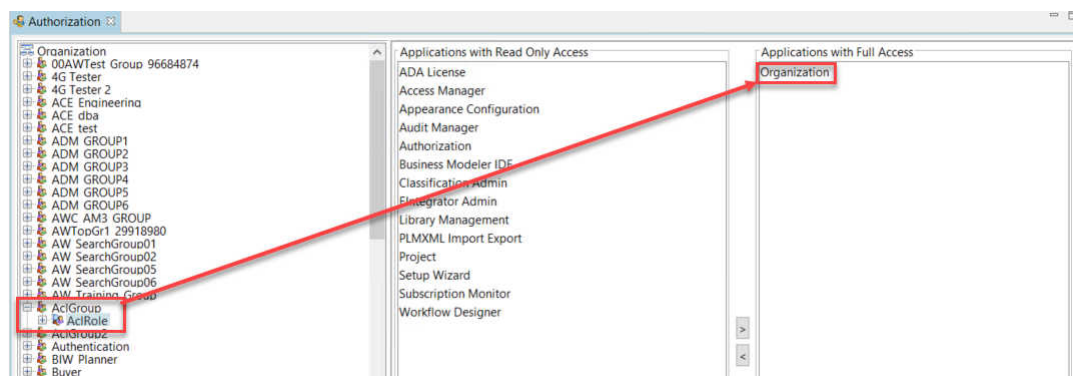
  3. To grant **Read**, **Write**, **Delete**, **Check-In/Check-Out**, and **Create** access for any non-dba user to access the Organization application, you must create an ACL.
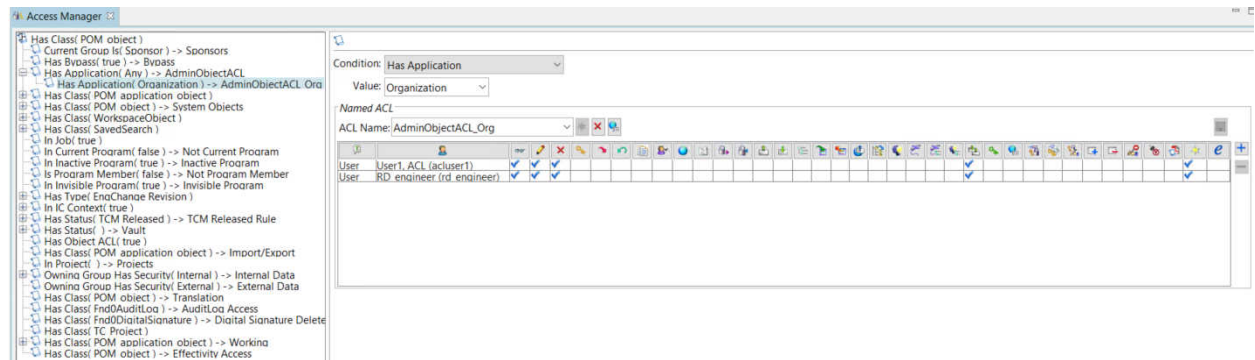
- **Example 2**
  As a customer, if you use the Authorization application to manage application security for the Access Manager and Authorization applications, then you must:

  1. Create an Access Manager rule condition in the Access Manager rule tree under the **Has Application(Any)** condition for the Access Manager and Authorization applications.

     - **Access Manager**
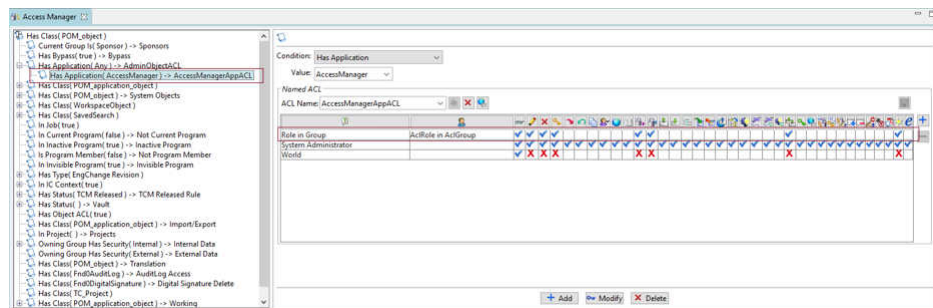
     - **Authorization**
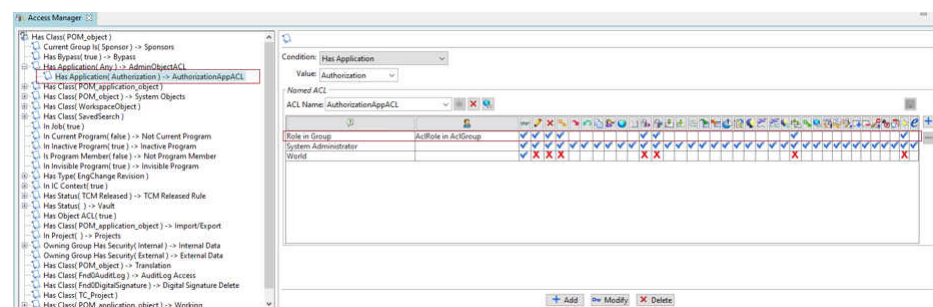
  2. Include the appropriate accessors in the ACL used against this condition.

> **Note:**
>
> You must make entries to the ACL for group and role accessors to make it like the accessors used for these applications in the Authorization application.

Has Attribute

CATEGORY

Default

DESCRIPTION

Specifies an attribute and value associated with a particular class. The given attribute should be a valid persistent attribute on the given class.

CONDITION EVALUATION

If the given attribute does not exist on the class, the rule tree evaluates to **false**.

INPUT ARGUMENTS

class:attribute=value

> Note:
>
> This condition supports the **!=** comparator. If **!=** is used with the **Has Attribute** rule tree condition, the condition evaluates to true if the value of the specified attribute on the object under evaluation is *not* equal to the value specified on the righthand side of the **!=** comparator. It will not support any other comparator like **<. >. <=,** or **>=**.

class	The class of the object for which you set the rule.
attribute	The attribute of the class. Supported attribute types include:

- **POM_string** (string)

- **POM_int** (integer)

- **POM_float** (float)

- **POM_logical** (logical)

- **POM_untyped_reference** (reference)

- **POM_external_reference** (reference)

- **POM_typed_reference** (reference)

value	The value for which the attribute is evaluated. *value* can contain wild cards.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- All subtypes of **POM_object**

EXAMPLE

The following shows how to use the **Has Attribute** condition with single-tag reference attributes, in this case, **owning_organization** and **owning_project**:

📖 Has Attribute (WorkspaceObject:owning_project=1) -> TestACL

📖 Has Attribute (Item:owning_organization=1) -> TestACL

The following example shows how to use the **Has Attribute** condition with a string attribute:

📖 Has Attribute(Item:object_name=test*)

The following example shows how to use the **Has Attribute** condition with a reference attribute:

📖 Has Attribute(Item:owning_organization=1)

- A value of **1** in the argument indicates that the condition expects the attribute value to be a **nonnull (nonzero)** value.

- A value of **0** in the argument indicates that the condition expects the attribute value to be a **null_tag** value.

- Do not use any string values. Only use **0** or **1**.

The following example shows how to use the **Has Attribute** condition with an integer attribute:

📖 Has Attribute(WorkspaceObject:revision_number=2)

BEST PRACTICES FOR RULES

- All the strings used in the rule tree are internal values.

- Blank spaces are not allowed in the rule syntax.

- Logical values must be either **0** (false) or **1** (true).

- References can only be checked for a **null_tag (0)** or **nonnull (nonzero)** value.

- **Has Attribute** supports only single value attributes. Attributes with variable-length arrays (VLAs) are not supported.

- **Has Attribute** does not support array attributes.

- **Has Attribute** supports the persistent attributes on the class.

- Do not use **Has Attribute** with compound properties or with types.

- Always use either the **Has IP Classification** rule or the **Has Government Classification** rule when **ip_classification** or **gov_classification** attributes are involved.

RELATED RULE CONDITIONS

- **Has Class**

- **Has Type**

- **Has Property**

Has Bypass

CATEGORY

Administrative

DESCRIPTION

Specifies whether the user has bypass privileges set. Bypass privilege supersedes other privileges.

CONDITION EVALUATION

true	If the user has bypass privileges, evaluates to **true**.
false	If the user does not have bypass privileges, evaluates to **false**.

INPUT ARGUMENTS

true or **false**

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Workspace objects

Has Class

CATEGORY

Default

DESCRIPTION

Specifies an object class. The object is evaluated to determine if it is of the specified class.

CONDITION EVALUATION

true	Evaluates to **true** if the object is of the specified class.
false	In all other cases, it evaluates to **false**.

INPUT ARGUMENTS

class-name

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

• Workspace objects

GOOD RULE PRACTICES

Do not use wildcard characters with the **Has Class** condition. For example, do not use **Has Class (Des*)**. **Has Class** requires full and correct class names.

RELATED RULE CONDITIONS

• **Has Attribute**

• **Has Type**

• **Has Property**

Has Classification

CATEGORY

General

DESCRIPTION

Validates the custom classification attribute value of the object against the value specified for the condition.

INPUT ARGUMENTS

Custom Classification Property Name{*operator*}Custom Classification attribute value

EXAMPLE

EAR_classification>=EAR_highest

RELATED RULE CONDITIONS

- **Has Government Classification**

- **Has IP Classification**

Has Description

CATEGORY

General

DESCRIPTION

Specifies a description for the object. The object is evaluated to determine whether the description matches this value.

CONDITION EVALUATION

true	Evaluates to **true** if the description of the object matches the specified description.
false	In all other cases, it evaluates to **false**.

INPUT ARGUMENTS

text-string   Text of the description to be evaluated.

> Note:
>
> The description value can contain wildcard characters.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

• Workspace objects

RELATED RULE CONDITIONS

• **Has Form Attribute**

• **Has Item ID**

• **Has Name**

Has Digital Signature

CATEGORY

General

DESCRIPTION

Specifies whether a business object has a digital signature of the specified status.

CONDITION EVALUATION

True   Evaluates to **True** if the attached digital signature has specified status.

False   In all other cases, it evaluates to **False**.

INPUT ARGUMENTS

Valid
Invalid
Propagated
Revoked
Voided

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

• **POM_APPLICATION_OBJECT** and its subtypes

> Note:
>
> This condition is installed only if the digital signature schema is installed.

RELATED RULE CONDITIONS

• **User Has Digital Signature**

Has Exclude License Of Category

CATEGORY

License by Category

DESCRIPTION

Checks whether the workspace object being evaluated has any exclude license of the given category.

CONDITION EVALUATION

true	If there is an exclude license with the specified category attached to the workspace object, evaluates to **true**.
false	If there is no exclude license with the specified category or if the license exists but is not attached to the workspace object, the condition evaluates to **false**.

INPUT ARGUMENTS

license_category   A string identifying the category of the license.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

• Any workspace object.

EXAMPLE

For an example, see Security Administration.

RELATED RULE CONDITIONS

• **Has ADA License Of Category**

• **Has IP License Of Category**

• **Has ITAR License Of Category**

Has Form Attribute

CATEGORY

General

DESCRIPTION

Enables access control of items and item revisions by setting conditions on attributes of the **Masterform** class. This rule can be applied to the **ItemRevisionMaster** form to control access to the item.

This rule can also be used to control write access to the properties of items and item revisions, which in turn determine who can add or remove datasets associated with the item or item revision through a **Specification** relation.

This rule cannot be used to control access to the datasets, and it cannot be applied to user-defined forms. It should be added below the **Working→Item Revision/Item Rule** rule in the rule tree.

> Note:
>
> The way Access Manager evaluates Master forms does not follow the normal rules. Master forms inherit access privileges from the parent item or item revision, so if you change access privileges to an item or item revision you affect the privileges on the Master form.
>
> You can use the **TC_MASTERFORM_DELEGATE** environment variable to change the default behavior.

INPUT ARGUMENTS

form-storage-class:attribute=value

form-storage-class	The storage class for the form type on which you set the rule.
attribute	The attribute of the form. Supported attribute types are **POM_string**, **POM_int**, and **POM_double**.
value	The value for which the attribute is evaluated.

> Note:
>
> Blank spaces are not allowed in the rule syntax.

RELATED RULE CONDITIONS

- **Has Description**

- **Has Item ID**

- **Has Name**

Has Government Classification

CATEGORY

International Traffic in Arms Regulations (ITAR)

DESCRIPTION

Compares the classification level in the condition argument with the object classification level. If the object is not classified, or if the object classification level is less than that of the given classification in the argument, this condition returns **True**.

INPUT ARGUMENTS

Classification levels (from the **ITAR_level_list_ordering**):

0   secret

1   top_secret,super_secret

EXAMPLE

When you have a rule, **'Has Government Classification (secret)'**, the code converts it to a security level of **0** and returns either **True** or **False** based on that.

- If more than one classification is on the same line in **ITAR_level_list_ordering**, each classification returns **1** and are equivalent.

- If each entry has a different line in **ITAR_level_list_ordering**, you can use **Has Government Classification** because each value would return a different level number.

Use **Has Attribute** to distinguish different classification entries on the same line in **ITAR_level_list_ordering**. For example:

```
Has Attribute(WorkspaceObject:gov_classification=secret) -> Secret ACL
Has Attribute(WorkspaceObject:gov_classification=top_secret)-> TopSecret ACL
Has Attribute(WorkspaceObject:gov_classification=super_secret)-> SuperSecret ACL
```

RELATED RULE CONDITIONS

- **User Has Government Clearance**

- **User Is Excluded**

- **User Is ITAR Licensed**

Has IP License Of Category

CATEGORY

License by Category

DESCRIPTION

Checks whether the workspace object being evaluated has any IP license of the given category.

CONDITION EVALUATION

true	If there is an IP license with the specified category attached to the workspace object, evaluates to **true**.
false	If there is no IP license with the specified category or if the license exists but is not attached to the workspace object, the condition evaluates to **false**.

INPUT ARGUMENTS

license_category   A string identifying the category of the license.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

• Workspace objects

EXAMPLE

For an example, see Security Administration.

RELATED RULE CONDITIONS

• **Has ADA License Of Category**

• **Has Exclude License Of Category**

• **Has ITAR License Of Category**

Has ITAR License Of Category

CATEGORY

License by Category

DESCRIPTION

Checks whether the workspace object being evaluated has any ITAR license of the given category.

CONDITION EVALUATION

true   If there is an ITAR license with the specified category attached to the workspace object, evaluates to **true**.

false   If there is no ITAR license with the specified category or if the license exists but is not attached to the workspace object, the condition evaluates to **false**.

INPUT ARGUMENTS

license_category   A string identifying the category of the license.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

• Workspace objects

EXAMPLE

For an example, see Security Administration.

RELATED RULE CONDITIONS

• **Has ADA License Of Category**

• **Has Exclude License Of Category**

• **Has IP License Of Category**

Has Item ID

CATEGORY

General

DESCRIPTION

Specifies an item ID against which the item is evaluated.

INPUT ARGUMENTS

item-id   This condition can only be used on **Item** objects.

> Note:
> - The item ID value can contain wildcard characters.
> - This condition can only be used on **Item** objects.

RELATED RULE CONDITIONS

- **Has Description**

- **Has Form Attribute**

- **Has Name**

Has Item Key

CATEGORY

General

DESCRIPTION

Specifies a multifield key identifier against which the item is evaluated. In a multifield key environment, multifield key identifiers are assigned to each object to ensure their uniqueness in the database.

For assistance obtaining the multifield key identifier defined for an item, use the following utilities:

- **get_key_definition**, which obtains the MFK definition for a class.

- **get_key_string**, which obtains the key string for an item.

CONDITION EVALUATION

true	If the item key ID matches the multifield key of the item, it evaluates to **true**.
false	In all other cases, it evaluates to **false**.

INPUT ARGUMENTS

item-key

Multifield key of the item.

> Note:
>
> The item key value can contain wildcard characters.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Item or item revision

EXAMPLE

You have a multifield key environment set up so that an item and its related objects have the same ID. You want to restrict access to the CAD data but allow access to the associated Word document. Set up the Access Manager rule as follows.

📘 Has Item Key (item_id=001,object_type=msword})

   📘 World –> Read
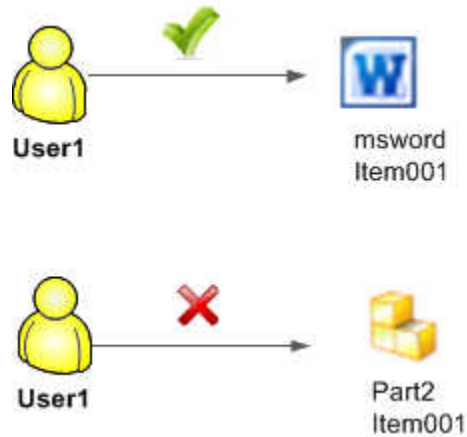
The rule states that a user is allowed access if the item has a multifield key ID of **{item_id=Item001,object_type=msword}**, with the **World** having read access.

RELATED RULE CONDITIONS

- **Has Item ID**

Has Metadata Class

CATEGORY

Administrative

DESCRIPTION

Provides additional security to property conditions and other metadata by granting all privileges to the **System Administrator** accessor and only allowing the **World** accessor to have **Read** access control list (ACL) privileges.
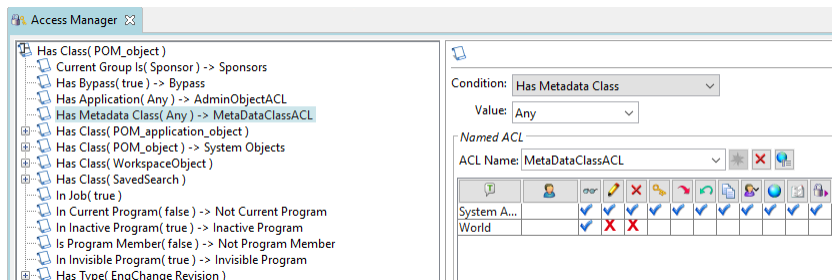
INPUT ARGUMENTS

Input arguments are metadata class names, for example:

> **Any**
> **BusinessRule**
> **Constant**
> **ConstantAttach**
> **ImanType**
> **POM_dd**
> **Property**

EXAMPLE

- As a customer, you want to restrict access by the **World** accessor to metadata and ensure only the **System Administrator** accessor has access to metadata.

Has Name

CATEGORY

General

DESCRIPTION

Specifies a name against which the object is evaluated.

CONDITION EVALUATION

true	Evaluates to **true** if the name value matches the specified name of the object evaluated.
false	In all other cases, it evaluates to **false**.

INPUT ARGUMENTS

text-string	Name value against which the object is evaluated.

> Note:
>
> The name value can contain wildcard characters.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Workspace objects

RELATED RULE CONDITIONS

- **Has Description**

- **Has Form Attribute**

- **Has Item ID**

Has Named Exclude License

CATEGORY

Licenses

DESCRIPTION

Checks whether the specified exclude license is attached to the workspace object being evaluated.

CONDITION EVALUATION

true	If there is an exclude license corresponding to the license ID and the license is attached to the workspace object, the condition evaluates to **true**.
false	In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

License ID	ID of the license to be attached to the workspace object.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Workspace objects

EXAMPLE

For an example, see **Has Named ITAR License**.

RELATED RULE CONDITIONS

- **Has Named IP License**

- **Has Named ITAR License**

- **Has Named ADA License**

Has Named IP License

CATEGORY

Licenses

DESCRIPTION

Checks whether a specific intellectual property (IP) license is attached to the workspace object being evaluated.

CONDITION EVALUATION

true	If there is an IP license corresponding to the license ID and the license is attached to the workspace object, the condition evaluates to **true**.
false	In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

License ID   The ID of the license to be attached to the workspace object.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

• Workspace objects

EXAMPLE

For an example, see **Has Named ITAR License**.

RELATED RULE CONDITIONS

• **Has Named Exclude License**

• **Has Named ITAR License**

• **Has Named ADA License**

Has Named ITAR License

CATEGORY

Licenses

DESCRIPTION

Checks whether the specified ITAR license is attached to the workspace object being evaluated.

CONDITION EVALUATION

true   If there is an ITAR license corresponding to the license ID and the license is attached to the workspace object, the condition evaluates to **true**.

false   In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

License ID   ID of the license to be attached to the workspace object.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

• Workspace objects

EXAMPLE

The following Access Manager rule states that a user is allowed access if there is an ITAR license by the name **ITAR001** attached to an object, with the **World** having read access:

📄 Has Named ITAR License (ITAR001)

   📄 World –> Read

User1 is allowed access because there is an ITAR license **ITAR001** attached to **Item001**, as shown next. However, **User1** is not allowed access to **Item002** because no **ITAR001** license is attached to it.
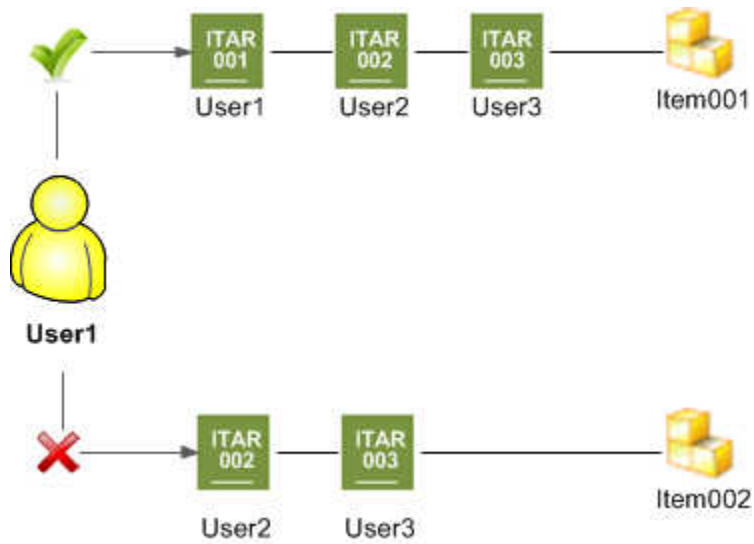
RELATED RULE CONDITIONS

- **Has Named Exclude License**

- **Has Named IP License**

- **Has Named ADA License**

Has Named ADA License

CATEGORY

Licenses

DESCRIPTION

Checks whether the specified ADA license is attached to the workspace object being evaluated.

CONDITION EVALUATION

true	If there is a license corresponding to the license ID and the license is attached to the workspace object, the condition evaluates to **true**.
false	In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

(Custom License:*LicenseID***)**	ID of the license to be attached to the workspace object.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

• Workspace objects

EXAMPLE

For an example, see **Has Named ITAR License**.

RELATED RULE CONDITIONS

• **Has Named Exclude License**

• **Has Named IP License**

• **Has Named ITAR License**

Has No Classification

CATEGORY

General

DESCRIPTION

Matches if the object has a null value for the custom classification attribute.

INPUT ARGUMENTS

Custom Classification Property Name

EXAMPLE

EAR_classification

RELATED RULE CONDITIONS

- **Has No Government Classification**

- **Has No IP Classification**

Has No Government Classification

CATEGORY

International Traffic in Arms Regulations (ITAR)

DESCRIPTION

Matches if the object has a null value for the government classification attribute.

RELATED RULE CONDITIONS

- **Has Government Classification**

Has No Status

CATEGORY

Default

DESCRIPTION

Supports the negation for the existing **Has Status** rule tree condition.

CONDITION EVALUATION

true	If the object under evaluation does not have the defined status, the condition evaluates to **true**.
false	In all other cases, the condition evaluates to **false**.

Has No IP Classification

CATEGORY

Intellectual property (IP)

DESCRIPTION

Checks whether the workspace object does not have a value specified in the IP classification attribute.

RELATED RULE CONDITIONS

- **User Has IP Clearance**

- **Has IP Classification**

- **User Is IP Licensed**

Has Object ACL

CATEGORY

Default

DESCRIPTION

Specifies that an ACL is associated with an object. This condition does not expect an ACL attached to a rule. It is a placeholder that indicates the point at which process ACLs and object ACLs are applied in the rule tree hierarchy.

INPUT ARGUMENTS

true or **false**

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

• Workspace objects

RELATED RULE CONDITIONS

• **In Job**

Has Property

CATEGORY

Default

DESCRIPTION

Specifies the value of a compound property against which an object is evaluated.

INPUT ARGUMENTS

The **Has Property** condition supports compound properties and persistent properties on the business object type. It supports multi-value (VLA) properties.

> Note:
>
> **Has Property** does not support the following property types:
>
> • Runtime
>
> • Relation
>
> • Table
>
> • Reference

Typename:prop_name=prop_value

> Note:
>
> This condition supports the **!=** comparator. If **!=** is used with the **Has Property** rule tree condition, the condition evaluates to true if the value of the specified attribute on the object under evaluation is *not* equal to the value specified on the righthand side of the **!=** comparator. It will not support any other comparator like **<. >. <=,** or **>=.**

Typename — The full object type.

prop_name — The name of a compound property on the business object.

prop_value — The value of the property against which the condition is evaluated. Supported property types include:

   • **PROP_string** (string) /**PROP_note** (short)

   • **PROP_char** (character)

- **PROP_int** (integer)

- **PROP_float** (float)

- **PROP_logical** (logical)

- **PROP_untyped_reference** (reference)

- **PROP_external_reference** (reference)

- **PROP_typed_reference** (reference)

> Note:
> - Property value can contain wild cards.
>
> - All the strings used in the rule tree are internal values.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Workspace objects

EXAMPLE

The following example shows how to use the **Has Property** condition with a string property:

📖 Has Property(Item:<string_prop_name> =test*)

The following example shows how to use the **Has Property** condition with a reference property:

📖 Has Property(Item:<reference_prop_name>=1)

- A value of **1** in the argument indicates that the condition expects the attribute value to be a **nonnull (nonzero)** value.

- A value of **0** in the argument indicates that the condition expects the attribute value to be a **null_tag** value.

The following example shows how to use the **Has Property** condition with a integer property:

Has Property(WorkspaceObject:<int_prop_name>=2)

The following example shows how to use the **Has Property** condition with a character property:

Has Property(WorkspaceObject:<char_prop_name>='c')

For an additional example of how to use the **Has Property** condition, see Security Administration.

RELATED RULE CONDITIONS

- **Has Attribute**

- **Has Class**

- **Has Type**

Has Status

CATEGORY

Default

DESCRIPTION

Specifies the status type against which the object is evaluated.

INPUT ARGUMENTS

status-name   Accepts null entry **null=all**.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

• Workspace objects

RELATED RULE CONDITIONS

• **Has Type**

Has Type

CATEGORY

Default

DESCRIPTION

Specifies the object type against which the object is evaluated.

INPUT ARGUMENTS

type-name	The full object type.

> Note:
>
> Do not use wildcard characters with the **Has Type** condition. For example, do not use **Has Type (Des*)**. **Has Type** requires full and correct type names.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

• Workspace objects

RELATED RULE CONDITIONS

• **Has Status**

In Current Program

CATEGORY

Program

DESCRIPTION

Specifies access based on whether the program to which the data is assigned is the current program under which the user is logged on to Teamcenter.

INPUT ARGUMENTS

true or **false**

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Workspace objects

EXAMPLE

For an example, see Security Administration.

RELATED RULE CONDITIONS

- **In Inactive Program**

- **In Invisible Program**

- **Is Owned By Program**

- **Is Program Member**

In IC Context

CATEGORY

Incremental Change

DESCRIPTION

Enables structure edits (occurrence edits, occurrence notes, transform edits, and attachment edits) to be controlled by the Structure Manager, Manufacturing Process Planner, Multi-Structure Manager, or Part Planner application. The rule does not depend on the properties of the object.

When there is an active incremental change in the structure editor, the **IC Context (true)** condition is satisfied and its associated ACL is applied.

INPUT ARGUMENTS

true or **false**

> Note:
>
> Always use the **true** value for this condition. The **false** value applies the rule to all objects, regardless of whether structure edits are being made.

In Inactive Program

CATEGORY

Program

DESCRIPTION

Controls access to data based on whether the status of the owning program is **inactive**.

INPUT ARGUMENTS

true or **false**

EXAMPLE

For an example, see Security Administration.

RELATED RULE CONDITIONS

- In Current Program

- In Invisible Program

In Invisible Program

CATEGORY

Program

DESCRIPTION

Controls access to data based on whether the status of the owning program is **invisible**.

INPUT ARGUMENTS

true or **false**

EXAMPLE

For an example, see Security Administration.

RELATED RULE CONDITIONS

- **In Current Program**

- **In Inactive Program**

- **Is Owned By Program**

- **Is Program Member**

In Job

CATEGORY

Default

DESCRIPTION

Specifies whether the target object is in a workflow job (process). This condition does not expect an ACL attached to a rule. It is a placeholder that indicates the point at which workflow ACLs are applied in the rule tree hierarchy.

> Note:
>
> No subbranches can be added below the **In Job** branch in the Access Manager rule tree.

INPUT ARGUMENTS

true or **false**

RELATED RULE CONDITIONS

- **Has Object ACL**

In Project

CATEGORY

Project

DESCRIPTION

Specifies a project to which the object must be assigned. The condition is evaluated as being true when the active project to which the object is assigned matches the project specified for this rule condition. If you use an empty string as the value for this condition, the condition is deemed true if the object is assigned to any active project.

INPUT ARGUMENTS

project-ID

The syntax for this rule is:

```
In Project (project-ID)-project_acl
```

EXAMPLE

For an example, see Security Administration.

RELATED RULE CONDITIONS

- **In Current Project**

- **Is Project Member**

Inactive Sequence

CATEGORY

General

DESCRIPTION

Specifies that previous sequences are historical and cannot be worked on independently. The latest sequence is always the working sequence for the revision.

> Note:
>
> This condition is used with the **Inactive Sequence Objects** ACL.

INPUT ARGUMENTS

true or **false**

IP License Has Citizenship

CATEGORY

License by Category

DESCRIPTION

Checks whether the IP license being evaluated has the given citizenship.

> Note:
>
> Citizenships are a two-letter country code from ISO 3166 (for example, Germany's country code is **DE**). A user can have multiple citizenships.

CONDITION EVALUATION

true	If any of the citizenships of the user being evaluated match the specified citizenship, the condition evaluates to **true**.
false	If none of the citizenships of the user being evaluated match the specified citizenship, the condition evaluates to **false**.

INPUT ARGUMENTS

citizenship	Two-character ISO 3166 codes identifying a country.
	This condition accepts negation using a minus (**–**) prefix. For example, **–IR** means that the user cannot have an IR citizenship.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Any workspace object.

Is Archived

CATEGORY

General

DESCRIPTION

> Note:
>
> This rule condition is implemented to support a legacy feature that is now obsolete. Siemens Digital Industries Software does not recommend this rule condition for new work.

Specifies that the object's archive status is evaluated.

INPUT ARGUMENTS

true or **false**

RELATED RULE CONDITIONS

- **Is Local**

Is Current Group External

CATEGORY

Ownership/Accessor based

DESCRIPTION

Evaluates whether the security of the current logged in group is external.

INPUT ARGUMENTS

true or **false**

EXAMPLE

For an example, see Security Administration.

RELATED RULE CONDITIONS

- **Is User External**

- **Is User In Current Group**

In Current Project

CATEGORY

Project

DESCRIPTION

Specifies the project ID against which the object is evaluated. The condition is evaluated as being true when the object is in the current active project of the logged-on user, and the project ID of the current project matches the value for this condition.

> Note:
>
> This rule is not delivered with the default installation of Teamcenter. It must be added manually.

INPUT ARGUMENTS

project-ID

The syntax for this rule is:

```
In Project (project-ID)-project_acl
```

EXAMPLE

For an example, see Security Administration.

RELATED RULE CONDITIONS

- **In Project**

- **Is Project Member**

Is GA

CATEGORY

Ownership/Accessor based

DESCRIPTION

Specifies whether the user's status as a group administrator in the current group is evaluated.

INPUT ARGUMENTS

true or **false**

RELATED RULE CONDITIONS

- **Is SA**

Is Local

CATEGORY

General

DESCRIPTION

Specifies whether the object's residence in the local database is evaluated. This condition is used when Multi-Site Collaboration is implemented.

INPUT ARGUMENTS

true or **false**

RELATED RULE CONDITIONS

* **Is Archived**

Is Group External

CATEGORY

Ownership/Accessor based

DESCRIPTION

Evaluates whether the object under consideration is a group object and has external security.

INPUT ARGUMENTS

true or **false**

EXAMPLE

For an example, see Security Administration.

RELATED RULE CONDITIONS

- **Is Current Group External**

- **Is Group Same As Current Group**

Is Group Member External

CATEGORY

Ownership/Accessor based

DESCRIPTION

Evaluates whether the object under consideration is GroupMember and belongs to a group that has external security.

INPUT ARGUMENTS

true or **false**

EXAMPLE

For an example, see Security Administration.

RELATED RULE CONDITIONS

- **Is Current Group External**

- **Is Member Group Same As Current Group**

Is Group Same As Current Group

CATEGORY

Ownership/Accessor based

DESCRIPTION

Evaluates whether the object under consideration is group and is same as the current logged in group.

INPUT ARGUMENTS

true or **false**

EXAMPLE

For an example, see Security Administration.

RELATED RULE CONDITIONS

- **Is Current Group External**

- **Is Group External**

Is Member Group Same As Current Group

CATEGORY

Ownership/Accessor based

DESCRIPTION

Evaluates whether the group member object belongs to the same group as the current logged on group.

INPUT ARGUMENTS

true or **false**

EXAMPLE

For an example, see Security Administration.

RELATED RULE CONDITIONS

- **Is Current Group External**

- **Is Group Member External**

Is Owned By Program

CATEGORY

Program

DESCRIPTION

Controls access to data based on whether data is owned by the program specified as a value for the **Is Owned By Program** condition.

INPUT ARGUMENTS

true or **false**

EXAMPLE

For an example, see Security Administration.

RELATED RULE CONDITIONS

- **In Current Program**

- **In Inactive Program**

- **In Invisible Program**

- **Is Program Member**

Is Program Member

CATEGORY

Program

DESCRIPTION

Specifies whether the user's membership in the program is evaluated.

> Note:
>
> This does not apply to project team members who are inactive group members.

CONDITION EVALUATION

true	Evaluates to **true** if the user is a member of the owning program or a shared program.
false	In all other cases, evaluates to **false**.

INPUT ARGUMENTS

true or **false**

EXAMPLE

For an example, see Security Administration.

RELATED RULE CONDITIONS

- **In Current Program**

- **In Inactive Program**

- **In Invisible Program**

Is Project Member

CATEGORY

Project

DESCRIPTION

Specifies whether the user's membership in the project is evaluated. This condition is only true when the user is a current member of the project.

INPUT ARGUMENTS

true or **false**

EXAMPLE

For an example, see Security Administration.

RELATED RULE CONDITIONS

- **In Current Project**

- **In Project**

- **Is Owned By Program**

Is User External

CATEGORY

Ownership/Accessor based

DESCRIPTION

Evaluates whether the user object is from a group whose security is external.

INPUT ARGUMENTS

true or **false**

EXAMPLE

For an example, see Security Administration.

RELATED RULE CONDITIONS

- **Is Current Group External**

- **Is User In Current Group**

Is User In Current Group

CATEGORY

Ownership/Accessor based

DESCRIPTION

Evaluates whether the user object under evaluation has current group membership.

INPUT ARGUMENTS

true or **false**

EXAMPLE

For an example, see Security Administration.

RELATED RULE CONDITIONS

- **Is Current User Group External**

- **Is User External**

Has Project Of Category

CATEGORY

Project

DESCRIPTION

Checks whether the workspace object being evaluated has any project assigned of the given category.

CONDITION EVALUATION

true	Evaluates to **true** if a project with the specified category is assigned to the workspace object.
false	In all other cases, evaluates to **false** if a project with the specified category is not assigned to the workspace object.

INPUT ARGUMENTS

project_category, which is a string identifying the category of the project.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

• Workspace objects

RELATED RULE CONDITIONS

• **In Current Project**

• **In Project**

• **Is Owned By Program**

Is SA

CATEGORY

Ownership/Accessor based

DESCRIPTION

Specifies whether the user's system administration group membership is evaluated.

INPUT ARGUMENTS

true or **false**

RELATED RULE CONDITIONS

- **Is GA**

Is Sponsored Mode

CATEGORY

General

DESCRIPTION

Checks whether the Teamcenter session is in sponsored mode. It enables end users to configure rules to enforce data access control when the Teamcenter session is launched in sponsored mode.

INPUT ARGUMENTS

true or **false**

RELATED RULE CONDITIONS

- **Current Group Is**

ITAR License Has Citizenship

CATEGORY

International Traffic in Arms Regulations (ITAR)

DESCRIPTION

Checks whether the ITAR license being evaluated has the given citizenship.

> Note:
>
> Citizenships are a two-letter country code from ISO 3166 (for example, Germany's country code is **DE**). A user can have multiple citizenships.

CONDITION EVALUATION

true   If any of the citizenships of the user being evaluated match the specified citizenship, the condition evaluates to **true**.

false   If none of the citizenships of the user being evaluated match the specified citizenship, the condition evaluates to **false**.

INPUT ARGUMENTS

citizenship   Two-character ISO 3166 codes identifying a country.

This condition accepts negation using a minus (**–**) prefix. For example, **–IR** means that the user cannot have an IR citizenship.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Any workspace object.

Has IP Classification

CATEGORY

Intellectual property (IP)

DESCRIPTION

Validates the IP classification attribute value of the object against the value specified for the condition.

The operators can be used without a clearance value; the IP classification attribute of the object is compared to the user's clearance level based on the specified operator.

> **Note:**
> - If the object has no IP classification attribute value, this rule does not apply.
>
> - This condition applies to an object that is IP classified, for example, **super-secret**. To set the IP classification to **super-secret**:
>
>   1. Select the object and choose **View→Properties**.
>
>   2. Check out the object.
>
>   3. Select **Show empty properties** and set **IP Classification** to **super-secret**.
>
>   4. Check in the object.

INPUT ARGUMENTS

Classification levels (from the **IP_level_list_ordering**):

0   secret

1   top_secret,super_secret

EXAMPLE

When you have a rule, **'Has IP Classification (secret)'**, the code converts it to a security level of **0** and returns either **True** or **False** based on that.

- If more than one classification is on the same line in **IP_level_list_ordering**, each classification returns **1** and are equivalent.

- If each entry has a different line in **IP_level_list_ordering**, you can use **Has IP Classification** because each value would return a different level number.

Use **Has Attribute** to distinguish different classification entries on the same line in **IP_level_list_ordering**. For example:

```
Has Attribute(WorkspaceObject:ip_classification=secret) -> Secret ACL
Has Attribute(WorkspaceObject:ip_classification=top_secret)-> TopSecret ACL
Has Attribute(WorkspaceObject:ip_classification=super_secret)-> SuperSecret ACL
```

RELATED RULE CONDITIONS

- **User Has IP Clearance**

- **Has No IP Classification**

- **User Is IP Licensed**

Owning Group

CATEGORY

Ownership/Accessor based

DESCRIPTION

Evaluates whether the object is owned by the group specified in the **group-name** argument.

INPUT ARGUMENTS

group-name

Wildcard characters can be used with the **Owning Group** condition to allow you to define rules applying to a group and all its subgroups. For example, assume that the **Design** group has two subgroups: **Analysis.Design** and **Development.Design**. By defining a value for the **Owning Group** condition using a wildcard, you can define a general rule to control access to all data owned by the **Design** group and its subgroups, for example:

📓 **Owning Group (*Design) –> design_group_acl**

EXAMPLE

For examples of managing group-level security, see Security Administration.

RELATED RULE CONDITIONS

- **Owning Group Has Security**

- **Owning Site**

- **Owning User**

Owning Group Has Security

CATEGORY

Ownership/Accessor based

DESCRIPTION

Evaluates whether the owning group of the object has a security string. This condition is true only if the security value of the owning group is equal to the value of this condition.

INPUT ARGUMENTS

Internal or **External**

EXAMPLE

For examples of managing group-level security, see Security Administration.

RELATED RULE CONDITIONS

- **Owning Group**

- **Owning Site**

- **Owning User**

Owning Site

CATEGORY

Ownership/Accessor based

DESCRIPTION

Evaluates whether the object is owned by the specified site. This condition is used when Multi-Site Collaboration is implemented.

INPUT ARGUMENTS

site-name

EXAMPLE

For examples of managing group-level security, see Security Administration.

RELATED RULE CONDITIONS

- **Owning Group**

- **Owning Group Has Security**

- **Owning User**

Owning User

CATEGORY

Ownership/Accessor based

DESCRIPTION

Evaluates whether the object is owned by the specified user.

INPUT ARGUMENTS

user-ID	ID of the user.

EXAMPLE

For examples of managing group-level security, see Security Administration

RELATED RULE CONDITIONS

- **Owning Group**

- **Owning Group Has Security**

- **Owning Site**

Site Geography

CATEGORY

International Traffic in Arms Regulations (ITAR)

DESCRIPTION

Checks whether the given geography matches the geography of the site being evaluated.

CONDITION EVALUATION

true	This condition evaluates to **true** if the given geography matches the geography of the site being evaluated.
false	In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

country-code	Two-character ISO 3166 country codes identifying the geography of the site.
	This condition accepts multiple geography arguments, which must be comma separated as shown in the Example section.
	Also, this condition accepts negation using a minus (**-**) prefix. For example, **-US** indicates any user at a site outside the U.S.
	When negating multiple geography arguments, you must enclose the geography arguments in parentheses. For example, **-(US,FR)** indicates the user is not authorized at a site in the U.S. and France.

EXAMPLE

- To specify the geography of a site in the U.S., enter **US**.

- To specify the geography of a site *outside* the U.S., enter **-US**.

- To specify the geography of a site in multiple countries (such as U.S., France, Japan, and India), enter **US,FR,JP,IN**.

- To specify the geography of a site *outside* multiple countries (such as U.S., France, Japan, and India), enter **-(US,FR,JP,IN)**.

RELATED RULE CONDITIONS

- **User Geography**

User-ADA Lic Has Citizenship

CATEGORY

Licenses

DESCRIPTION

Checks whether the user's citizenship matches the passed-in value and then checks if the user's citizenship is listed on any of the ADA licenses attached to the workspace object being evaluated.

CONDITION EVALUATION

true	This condition evaluates to **true** if the user's citizenship matches the input citizenship and that citizenship is listed on any nonexpired ADA license attached to the workspace object.
false	In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

(Custom License:citizenship**)**	Two-character ISO 3166 codes identifying a country. This condition accepts negation using a minus (–) prefix. For example, **–IR** means that the user cannot have an IR citizenship.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

• Any workspace object.

RELATED RULE CONDITIONS

• **User-Exclude Lic Has Citizenship**

• **User-IP Lic Has Citizenship**

• **User-ITAR Lic Has Citizenship**

User Citizenship

CATEGORY

International Traffic in Arms Regulations (ITAR)

DESCRIPTION

Checks whether the given citizenship matches the citizenships of the user being evaluated.

CONDITION EVALUATION

true	This condition evaluates to **true** if the given citizenships match the citizenships of the user being evaluated.
false	In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

citizenship	Two-character ISO 3166 country codes identifying citizenship of a user.
	This condition accepts multiple citizenship arguments, which must be comma separated as shown in the Example section.
	Also, this condition accepts negation using a minus (**-**) prefix. For example, **-US** indicates the user cannot have U.S. citizenship.
	When negating multiple citizenship arguments, you must enclose the citizenship arguments in parentheses. For example, **-(US,FR)** indicates the user cannot have citizenship in the U.S. and France.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

• Any workspace object.

EXAMPLE

• To specify a user having U.S. citizenship, enter **US**.

• To specify a user not having U.S. citizenship, enter **-US**.

• To specify a user having citizenship in multiple countries (such as U.S., France, Japan, and India), enter **US,FR,JP,IN**.

- To specify a user not having citizenship in multiple countries (such as U.S., France, Japan, and India), enter **-(US,FR,JP,IN)**.

RELATED RULE CONDITIONS

- **User Citizenship Or Nationality**

- **User Nationality**

User Citizenship Or Nationality

CATEGORY

International Traffic in Arms Regulations (ITAR)

DESCRIPTION

Checks whether the given citizenship matches the citizenship or nationality of the user being evaluated.

CONDITION EVALUATION

true	This condition evaluates to **true** if the citizenship matches the citizenship or nationality of the user being evaluated.
false	In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

citizenship	Two-character ISO 3166 country codes identifying the citizenship of the user.
	This condition accepts multiple citizenship arguments, which must be comma separated as shown in the Example section.
	Also, this condition accepts negation using a minus (**-**) prefix. For example, **-US** indicates the user cannot have U.S. citizenship.
	When negating multiple citizenship arguments, you must enclose the citizenship arguments in parentheses. For example, **-(US,FR)** indicates the user cannot have citizenship in the U.S. and France.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Any workspace object.

EXAMPLE

- To specify a user having U.S. citizenship, enter **US**.

- To specify a user not having U.S. citizenship, enter **-US**.

- To specify a user having citizenship in multiple countries (such as U.S., France, Japan, and India), enter **US,FR,JP,IN**.

- To specify a user not having citizenship in multiple countries (such as U.S., France, Japan, and India), enter **-(US,FR,JP,IN)**.

RELATED RULE CONDITIONS

- **User Citizenship**

- **User Nationality**

User-Exclude Lic Has Citizenship

CATEGORY

Licenses

DESCRIPTION

Checks whether the user's citizenship matches the passed-in value and then checks if the user's citizenship is listed on any of the exclude licenses attached to the workspace object being evaluated.

CONDITION EVALUATION

true	This condition evaluates to **true** if the user's citizenship matches the input citizenship and that citizenship is listed on any nonexpired exclude license attached to the workspace object.
false	In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

citizenship	Two-character ISO 3166 codes identifying a country.
	This condition accepts negation using a minus (**–**) prefix. For example, **–IR** means that the user cannot have an IR citizenship.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Any workspace object.

RELATED RULE CONDITIONS

- **User-ADA Lic Has Citizenship**

- **User-IP Lic Has Citizenship**

- **User-ITAR Lic Has Citizenship**

User Has Clearance

CATEGORY

General

DESCRIPTION

Validates the user's custom clearance level (from the attached custom LOV) against the value specified for the condition's input argument.

INPUT ARGUMENTS

Custom Clearance Property Name {*operator*} Custom Classification attribute value

EXAMPLE

EAR_clear>=EAR_highest

RELATED RULE CONDITIONS

- **User Has Government Clearance**

- **User Has IP Clearance**

User Has Digital Signature

CATEGORY

General

DESCRIPTION

Specifies whether a particular business object has a digital signature of the specified status in the context of the logged-in user.

CONDITION EVALUATION

True	Evaluates to **True** if the attached digital signature has specified status in the context of the logged-on user.
False	In all other cases, it evaluates to **False**.

INPUT ARGUMENTS

> **Valid**
> **Invalid**
> **Propagated**
> **Revoked**
> **Voided**

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- **POM_APPLICATION_OBJECT** and its subtypes

> Note:
>
> This condition is installed only if the digital signature schema is installed.

RELATED RULE CONDITIONS

- **Has Digital Signature**

User Has Government Clearance

CATEGORY

International Traffic in Arms Regulations (ITAR)

DESCRIPTION

Validates the user's government clearance level (**secret**, **super-secret**, **top-secret**) against the value specified for the condition's input argument.

Teamcenter defines out-of-the-box clearance levels using the **ITAR_level_list_ordering** preference as **secret**, **super-secret**, **top-secret**. This list can be customized.

This condition has two modes of evaluation:

- If the input argument specifies an operator and a clearance value, the condition compares this input value to the user's government clearance.

  Example: HasGovernmentClearance (>Secret)

- The operators can be used without a clearance value, in which case the user's government clearance is compared to the government classification attribute of the object based on the specified operator.

  Example: HasGovernmentClearance (>)

> Note:
>
> If the object is not ITAR classified (**gov_classification** attribute value is empty), the **User Has Government Clearance** condition always evaluates as being **true** regardless of whether or not the user is assigned a government clearance level.

CONDITION EVALUATION

true Evaluates to **true** in the following scenarios:

- The workspace object being evaluated does not have government classification set on it. Therefore, this evaluates to **true** because the data is not classified, and the user's clearance does not have any effect.
  Example:
  HasGovernmentClassification()

User's Gov Classification	Object's Gov Classification	Evaluation
		True
secret		True

- The condition has an input argument value and the user's government clearance value matches the condition's input argument.
  Example:
  HasGovernmentClassification(>secret)

User's Gov Classification	Evaluation
top-secret	True
secret	False

- The condition's input argument contains only an operator (without a clearance value), and the user's government clearance level matches the object's government classification attribute.

- The condition has no input argument, and the user's government clearance level is greater than or equal to the object's government classification level.
  Example:
  HasGovernmentClassification()

User's Gov Classification	Object's Gov Classification	Evaluation
top-secret	secret	True
secret	top-secret	False

- The user's government clearance level is not set, the object's government classification level is not set, and the government clearance value is specified for the condition as follows:

  >
  <
  =
  ==
  >=
  <=

false Evaluates to **false** in all other cases, including the case where the object being evaluated is not a subtype of **WorkspaceObject**.

INPUT ARGUMENTS

clearance_val ue Specific government clearance attribute values that can be prefixed by the following operators:

> >
> >=
> <
> <=
> =

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Workspace objects

EXAMPLE

The following example shows how to use the **User Has Government Clearance** condition using operators and a clearance value:

📘 User Has Government Clearance (>=secret) -> TestACL

📘 User Has Government Clearance (=topsecret) -> TestACL

The following example shows how to use the **User Has Government Clearance** condition using an operator without a clearance value:

📘 User Has Government Clearance (>=) -> TestACL

The following example shows how to use the **User Has Government Clearance** condition without any value for the condition:

📘 User Has Government Clearance () -> TestACL

RELATED RULE CONDITIONS

- **Has Government Classification**

- **Has No Government Classification**

- **User Is Excluded**

- **User Is ITAR Licensed**

User Has IP Clearance

CATEGORY

Intellectual property (IP)

DESCRIPTION

Validates the user's clearance level against the value specified for the condition.

The Intellectual property (IP) clearance level is the level of access the user has to sensitive (classified) information.

The operators can be used without a clearance value in which case the user's clearance is compared to the IP classification attribute of the object based on the specified operator.

> Note:
>
> If the data is not IP classified, the **User Has IP Clearance** condition is evaluated as being true regardless of whether or not the user is assigned a clearance level.

CONDITION EVALUATION

true   Evaluates to **true** in the following scenarios:

- The workspace object being evaluated does not have IP classification set on it.

- The condition has a clearance value specified and the user's IP clearance level matches the value specified for the condition.

- Operators are specified without a clearance value and the user's IP clearance level matches the IP classification specified on the object being evaluated, based on the specified operator.

- The IP clearance value is not specified for the condition, and the user's IP clearance level is greater than or equal to the object's IP classification level.
  Example:
  User Has IP Clearance (>=secret) -> TestACL

User's IP Clearance	Evaluation
top-secret	True
secret	True

- The IP clearance value is specified as **"="/">="/"<="** for the condition, the user's IP clearance level is not set, and the object's IP classification level is not set.

false   Evaluates to **false** in all other cases, including the case where the object being evaluated is not a subtype of **WorkspaceObject**.

INPUT ARGUMENTS

clearance_value   Specific IP clearance values that can be prefixed by the following operators:

> **>**

> **>=**

> **<**

> **<=**

> **=**

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Workspace objects

EXAMPLE

The following example shows how to use the **User Has IP Clearance** condition using operators and a clearance value:

📓 User Has IP Clearance (>=secret) -> TestACL

📓 User Has IP Clearance (=topsecret) -> TestACL

The following example shows how to use the **User Has IP Clearance** condition using an operator without a clearance value:

📓 User Has IP Clearance (>=) -> TestACL

The following example shows how to use the **User Has IP Clearance** condition without any value for the condition:

User Has IP Clearance () -> TestACL

RELATED RULE CONDITIONS

- **Has IP Classification**

- **Has No IP Classification**

- **User Is IP Licensed**

User In Attach ADA Lic of Ctgry

CATEGORY

Intellectual Property (IP)

DESCRIPTION

Checks the following:

- Whether the evaluation object is a workspace object (**WorkspaceObject**) or one of its subtypes.

- The workspace object has ADA licenses attached that:

  - Match the license category of the input category.

  - List the current session user on the license.

CONDITION EVALUATION

true   Evaluates to **true** if:

- The evaluation object a workspace object or one of its subtypes.

- The workspace object has ADA licenses attached that:

  - Match the license category of the input category.

  - List the current session user on the license.

false   Evaluates to **false** if:

- The evaluation object is not a workspace object.

- The input category is a wildcard character (*, %, @).

- The workspace object has no ADA licenses attached.

- If ADA licenses are attached, none of them list both the user and match the category.

INPUT ARGUMENTS

(Custom License:_license_category_**)**   A string identifying the name of the license category.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Any workspace object.
  If the evaluated object is not a workspace object, the condition returns **false**.

EXAMPLE

For an example, see Security Administration.

GOOD RULE PRACTICES

Access control by licenses can be configured based on the license type to vary access at a high level or based on the license name to vary the access at a granular level. Categories offer a way to control access by licenses in between the high and granular levels. They provide a way to have different subtypes of licenses under each type and configure access based on each category.

To learn more about categories, see Security Administration.

RELATED RULE CONDITIONS

- **User In Attach Excl Lic of Ctgry**

- **User In Attach IP Lic of Ctgry**

- **User In Attach ITAR Lic of Ctgry**

- **User In Attached Exclude License**

- **User In Attached IP License**

- **User In Attached License**

- **User In Attached ITAR License**

User In Attach Excl Lic of Ctgry

CATEGORY

Intellectual Property (IP)

DESCRIPTION

Checks the following:

- Whether the evaluation object is a workspace object (**WorkspaceObject**) or one of its subtypes.

- The workspace object has exclude licenses attached that:

  - Match the license category of the input category.

  - List the current session user on the license.

CONDITION EVALUATION

true   Evaluates to **true** if:

- The evaluation object is a workspace object or one of its subtypes.

- The workspace object has ITAR licenses attached that:

  - Match the license category of the input category.

  - List the current session user on the license.

false   Evaluates to **false** if:

- The evaluation object is not a workspace object.

- The input category is a wildcard character (*, %, @).

- The workspace object has no exclude licenses attached.

- If exclude licenses are attached, none of them list both the user and match the category.

INPUT ARGUMENTS

license_category   A string identifying the name of the license category.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Any workspace object.
  If the evaluated object is not a workspace object, the condition returns **false**.

EXAMPLE

For an example, see Security Administration.

GOOD RULE PRACTICES

Access control by licenses can be configured based on the license type to vary access at a high level or based on the license name to vary the access at a granular level. Categories offer a way to control access by licenses in between the high and granular levels. They provide a way to have different subtypes of licenses under each type and configure access based on each category.

To learn more about categories, see Security Administration.

RELATED RULE CONDITIONS

- **User In Attach ADA Lic of Ctgry**

- **User In Attach IP Lic of Ctgry**

- **User In Attach ITAR Lic of Ctgry**

- **User In Attached Exclude License**

- **User In Attached IP License**

- **User In Attached License**

- **User In Attached ITAR License**

User In Attach IP Lic of Ctgry

CATEGORY

Intellectual Property (IP)

DESCRIPTION

Checks the following:

- Whether the evaluation object is a workspace object (**WorkspaceObject**) or one of its subtypes

- The workspace object has IP licenses attached that:

    - Match the license category of the category input.

    - List the current session user on the license.

CONDITION EVALUATION

true   Evaluates to **true** if:

- The evaluation object is a workspace object or one of its subtypes.

- The workspace object has any IP licenses attached that:

    - Match the license category of the input category.

    - List the current session user on the license.

false   Evaluates to **false** if:

- The evaluation object is not a workspace object.

- The input category is a wildcard character (*, %, @).

- The workspace object has no IP licenses attached.

- If ITAR licenses are attached, none of them list both the user and match the category.

INPUT ARGUMENTS

license_category   A string identifying the name of the license category.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Any workspace object.
  If the evaluated object is not a workspace object, the condition returns **false**.

EXAMPLE

For an example, see Security Administration.

GOOD RULE PRACTICES

Access control by licenses can be configured based on the license type to vary access at a high level or based on the license name to vary the access at a granular level. Categories offer a way to control access by licenses in between the high and granular levels. They provide a way to have different subtypes of licenses under each type and configure access based on each category.

To learn more about license categories, see Security Administration.

RELATED RULE CONDITIONS

- **User In Attach ADA Lic of Ctgry**

- **User In Attach Excl Lic of Ctgry**

- **User In Attach ITAR Lic of Ctgry**

- **User In Attached Exclude License**

- **User In Attached IP License**

- **User In Attached License**

- **User In Attached ITAR License**

User In Attach ITAR Lic of Ctgry

CATEGORY

International Traffic in Arms Regulations (ITAR)

DESCRIPTION

Checks the following:

- Whether the evaluation object is a workspace object (**Workspace Object**) or one of its subtypes.

- Whether the workspace object has ITAR licenses attached that:

  - Match the license category of the input category.

  - List the current session user on the license.

CONDITION EVALUATION

true   Evaluates to **true** if:

- The evaluation object is a workspace object or one of its subtypes.

- The workspace object has ITAR licenses attached that:

  - Match the license category of the input category.

  - List the current session user on the license.

false   Evaluates to **false** if:

- The evaluation object is not a workspace object.

- The input category is a wildcard character (*, %, @).

- The workspace object has no ITAR licenses attached.

- If ITAR licenses are attached, none of them list both the user and match the category.

INPUT ARGUMENTS

license_category   A string identifying the name of the license category.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Any workspace object.
  If the evaluated object is not a workspace object, the condition returns **false**.

EXAMPLE

For an example, see Security Administration.

GOOD RULE PRACTICES

Access control by licenses can be configured based on the license type to vary access at a high level or based on the license name to vary the access at a granular level. Categories offer a way to control access by licenses in between the high and granular levels. They provide a way to have different subtypes of licenses under each type and configure access based on each category.

To learn more about license categories, see Security Administration.

RELATED RULE CONDITIONS

- **User In Attach ADA Lic of Ctgry**

- **User In Attach Excl Lic of Ctgry**

- **User In Attach IP Lic of Ctgry**

- **User In Attached Exclude License**

- **User In Attached IP License**

- **User In Attached License**

- **User In Attached ITAR License**

User In Attached ADA License

CATEGORY

Licenses

DESCRIPTION

Checks whether the user from the current session is listed on any or all of the custom licenses attached to the workspace object being evaluated.

CONDITION EVALUATION

true	Evaluates to **true** if:

- If set to **Any**, the condition evaluates to **true** if the user is listed on at least one custom license attached to the workspace object.

- If set to **All**, the condition evaluates to **true** if the user is listed on all custom licenses attached to the workspace object.

false	In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

- **Any**

- **All**

- (Custom License:{**Any|All|None**})

EXAMPLE

EAR_itarlicense:Any.

RELATED RULE CONDITIONS

- **User In Attached IP License**

- **User In Attached ITAR License**

- **User In Attached Exclude License**

User In Attached Exclude License

CATEGORY

Licenses

DESCRIPTION

Checks whether the user from the current session is listed in any or all exclude licenses attached to the workspace object being evaluated.

CONDITION EVALUATION

true	Evaluates to **true** if:

- If set to **Any**, the condition evaluates to **true** if the user is listed on any nonexpired exclude licenses attached to the workspace object.

- If set to **All**, the condition evaluates to **true** if the user is listed on all nonexpired exclude licenses attached to the workspace object.

- If set to **None**, the condition evaluates to **true** if the user is *not* listed in any of the attached Exclude licenses on the object under evaluation.

false	In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

Any or **All**

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Workspace objects

EXAMPLE

For an example, see **User In Attached ITAR License**.

RELATED RULE CONDITIONS

- **User In Attached IP License**

- **User In Attached ITAR License**

- **User In Attached License**

User In Attached IP License

CATEGORY

Licenses

DESCRIPTION

Checks whether the user being evaluated is listed on any or all of the IP licenses attached to the workspace objects.

CONDITION EVALUATION

true   Evaluates to **true** if:

- If set to **Any**, the condition evaluates to **true** if the user is listed on at least one nonexpired IP license attached to the workspace object.

- If set to **All**, the condition evaluates to **true** if the user is listed on all nonexpired IP licenses attached to the workspace object.

- If set to **None**, the condition evaluates to **true** if the user is *not* listed in any of the attached IP licenses on the object under evaluation.

false   In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

Any or **All**

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Workspace objects

EXAMPLE

For an example, see **User In Attached ITAR License**.

RELATED RULE CONDITIONS

- **User In Attached Exclude License**

- **User In Attached ITAR License**

- **User In Attached License**

User In Attached ITAR License

CATEGORY

Licenses

DESCRIPTION

Checks whether the user from the current session is listed on any or all of the ITAR licenses attached to the workspace object being evaluated.

CONDITION EVALUATION

true   Evaluates to **true** if:

- If set to **Any**, the condition evaluates to **true** if the user is listed on any nonexpired ITAR license attached to the workspace object.

- If set to **All**, the condition evaluates to **true** if the user is listed on all nonexpired ITAR licenses attached to the workspace object.

- If set to **None**, the condition evaluates to **true** if the user is *not* listed in any of the attached ITAR licenses on the object under evaluation.

false   In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

Any or **All**

In all other cases, the condition evaluates to **false**.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Workspace objects

EXAMPLE

The following Access Manager rule states that a user only needs to be on one or more of the ITAR licenses attached to an object to be given access to that object, with **World** having read access:

📁 User in Attached ITAR License (Any)

   📁 World –> Read

User1 is listed on one of the licenses attached to **Item001**, as shown. Therefore, **User1** is allowed access to **Item001**. **User5**, on the other hand, is not listed on any of the ITAR licenses attached to **item002** so **User5** is not given access to **item002**.
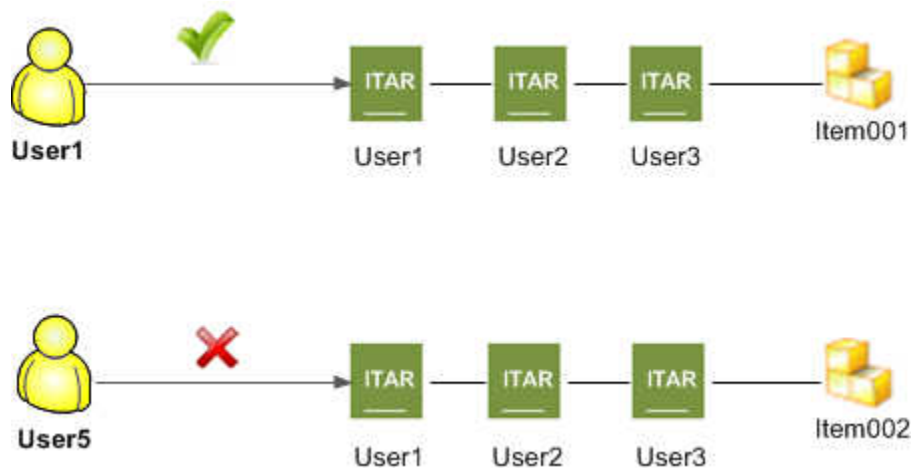
RELATED RULE CONDITIONS

- **User In Attached Exclude License**

- **User In Attached IP License**

- **User In Attached License**

User In Attached License

CATEGORY

Licenses

DESCRIPTION

Checks whether the user from the current session is listed on any or all of the licenses attached to the workspace object being evaluated.

CONDITION EVALUATION

true   Evaluates to **true** if:

- If set to **Any**, the condition evaluates to **true** if the user is listed on any nonexpired ADA license attached to the workspace object.

- If set to **All**, the condition evaluates to **true** if the user is listed on all nonexpired ADA licenses attached to the workspace object.

- If set to **None**, the condition evaluates to **true** if the user is *not* listed in any of the attached licenses on the object under evaluation.

false   In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

Any or **All**

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Workspace objects

EXAMPLE

For an example, see **User In Attached ITAR License**.

RELATED RULE CONDITIONS

- **User In Attached Exclude License**

- **User In Attached IP License**

- **User In Attached ITAR License**

User In License

CATEGORY

ADA

DESCRIPTION

Checks whether the **ADA_License** object being evaluated lists the user being evaluated, either individually or as a member of a group, so you can control the licenses that are visible to the user in Teamcenter applications, such as when searching for licenses, viewing licenses in the ADA License application, attaching licenses to an object, or viewing licenses attached to an object. For example, it determines whether Teamcenter displays a particular license in the **ADA licenses** view to the user, as shown, or in the **Attach an object to Licenses** dialog box.

CONDITION EVALUATION

true
- If set to **true**, the condition returns **true** if the user being evaluated is listed on the license either individually or as a member of a group.

- If set to **false**, the condition returns **true** if the user being evaluated is *not* listed on the license either individually or as a member of a group.

false
- If set to **true**, the condition returns **false** if the user being evaluated is *not* listed on the license either individually or as a member of a group.

- If set to **false**, the condition returns **false** if the user being evaluated is listed on the license either individually or as a member of a group.

INPUT ARGUMENTS

true or **false**

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- **ADA_License** object or any of its subclasses (**ITAR_License**, **IP_License**, or **Exclude_License**)

User In Named ADA License

CATEGORY

Licenses

DESCRIPTION

Checks whether the user being evaluated is listed on a custom license of the specified name. It does not check if the license is attached to the workspace objects being evaluated.

INPUT ARGUMENTS

Custom License: *License ID*

EXAMPLE

EAR_itarlicense:ear_license_01

RELATED RULE CONDITIONS

- **User In Named IP License**

- **User In Named ITAR License**

- **User In Named Exclude License**

User In Named Exclude License

CATEGORY

Licenses

DESCRIPTION

Checks whether a user being evaluated is listed in an exclude license of the specified license ID. It does not check if the license is attached to the workspace object being evaluated.

CONDITION EVALUATION

true   If the user is in the specified license and the license is an exclude license, the rule condition evaluates to **true**, regardless of whether the license is attached to the workspace object.

false   In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

License ID   ID of the license.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Workspace objects

EXAMPLE

For an example, see **User In Named ITAR License**.

RELATED RULE CONDITIONS

- **User In Named IP License**

- **User In Named ITAR License**

- **User In Named License**

User In Named IP License

CATEGORY

Licenses

DESCRIPTION

Checks whether the user being evaluated is listed on an IP license of the specified name. It does not check if the license is attached to the workspace objects being evaluated.

CONDITION EVALUATION

true   If the user is in the specified license and the license is an IP license, the rule condition evaluates to **true**, regardless of whether the license is attached to the workspace object.

false   In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

License ID   ID of the license.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

• Workspace objects

EXAMPLE

For an example, see **User In Named ITAR License**.

RELATED RULE CONDITIONS

• **User In Named Exclude License**

• **User In Named ITAR License**

• **User In Named License**

User In Named ITAR License

CATEGORY

Licenses

DESCRIPTION

Checks whether the user being evaluated is listed on an ITAR license of the specified name. It does not check if the license is attached to the workspace objects being evaluated.

CONDITION EVALUATION

true   If the user is in the specified license and the license is an ITAR license, the rule condition evaluates to **true**, regardless of whether the license is attached to the workspace object.

false   In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

License ID   ID of the license.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

• Workspace objects

EXAMPLE

The following Access Manager rules states that a user must be in a named ITAR license to be given access to an object, with the **World** having read access:

📘 Has GovClassification = Secret

📘 User In Named ITAR License (ITAR001)

   📘 World –> Read

The **ITAR 001** license has three users named on it (**User 1**, **User 2**, and **User 3**). In addition, the item trying to be accessed, **item001**, has a **gov_classification** set to **secret**.

Using the **User In Named ITAR license** condition, **User 1** can read **item001** because **User 1** is listed on the license, while **User 4** cannot read **item001** because **User 4** is not listed on the license.
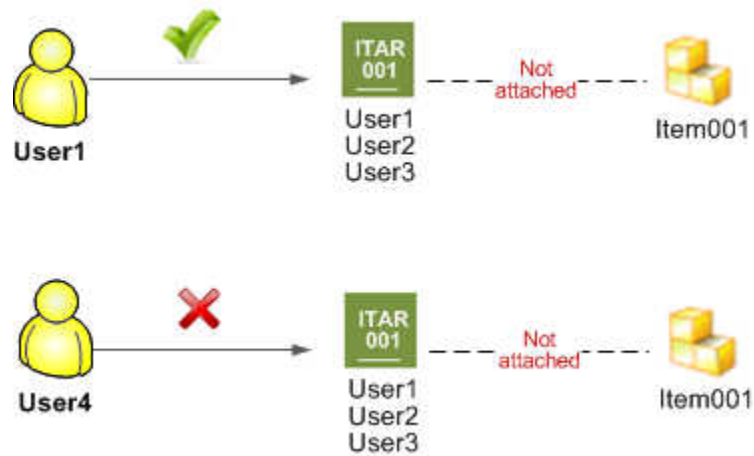
RELATED RULE CONDITIONS

- **User In Named Exclude License**

- **User In Named IP License**

- **User In Named License**

User In Named License

CATEGORY

Licenses

DESCRIPTION

Checks whether a user from the current session is listed in the license of the specified license ID. The rule condition does not check if the license is attached to the workspace object being evaluated.

CONDITION EVALUATION

true	If the user is in the specified license, the rule condition evaluates to **true**, regardless of whether the license is attached to the workspace object.
false	In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

License ID	ID of the license.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Workspace objects

EXAMPLE

For an example, see **User In Named ITAR License**.

RELATED RULE CONDITIONS

- **User In Named Exclude License**

- **User In Named IP License**

- **User In Named ITAR License**

User-IP Lic Has Citizenship

CATEGORY

Licenses

DESCRIPTION

Checks whether the user's citizenship matches the passed-in value and then checks if the user's citizenship is listed on any of the IP licenses attached to the workspace object being evaluated.

CONDITION EVALUATION

true	This condition evaluates to **true** if the user's citizenship matches the input citizenship and that citizenship is listed on any nonexpired IP license attached to the workspace object.
false	In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

citizenship	Two-character ISO 3166 codes identifying a country.
	This condition accepts negation using a minus (**–**) prefix. For example, **–IR** means that the user cannot have an IR citizenship.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Any workspace object.

RELATED RULE CONDITIONS

- **User-ADA Lic Has Citizenship**

- **User-Exclude Lic Has Citizenship**

- **User-ITAR Lic Has Citizenship**

User Is ADA Licensed

CATEGORY

General

DESCRIPTION

Checks whether the user currently logged on is cited in a valid (not expired) custom license attached to the workspace object either directly or by membership in a cited organization (group).

CONDITION EVALUATION

true
- If set to **true**, the condition returns **true** if the user being evaluated is cited in any valid (not expired) ADA license attached to the workspace object being evaluated either directly or as a member of a group.

- If set to **false**, the condition returns **true** if the user being evaluated is *not* cited in any valid (not expired) ADA license attached to the workplace object being evaluated either directly or as a member of a group.

false
- If set to **true**, the condition returns **false** if the user being evaluated is *not* listed in any valid (not expired) ADA license attached to the workspace object being evaluated either individually or as a member of a group.

- If set to **false**, the condition returns **false** if the user being evaluated is listed in any valid (not expired) ADA license attached to the workspace object being evaluated either individually or as a member of a group.

INPUT ARGUMENTS

Custom License:{**true|false**}

EXAMPLE

EAR_itarlicense:true

RELATED RULE CONDITIONS

- **User Is IP Licensed**

- **User Is ITAR Licensed**

User Is Excluded

CATEGORY

International Traffic in Arms Regulations (ITAR)

DESCRIPTION

Tests whether the user is cited in a valid (not expired) exclude license attached to the workspace object either directly or by membership in a cited organization (group).

CONDITION EVALUATION

true
- If the input argument is set to **true**, the condition evaluates to **true** if the user is cited in any valid (not expired) exclude license attached to the workspace object being evaluated either directly or by membership in a cited organization (group).

- If the input argument is set to **false**, the condition evaluates to **true** if the user is not cited in any valid (not expired) exclude license attached to the workspace object being evaluated either directly or by membership in a cited organization (group).

false
- If the input argument is set to **true**, the condition evaluates to **false** if the user is not cited in any valid (not expired) exclude license attached to the workspace object being evaluated either directly or by membership in a cited organization (group).

- If the input argument is set to **false**, the condition evaluates to **false** if the user is cited in any valid (not expired) exclude license attached to the workspace object being evaluated either directly or by membership in a cited organization (group).

INPUT ARGUMENTS

true or **false**

RELATED RULE CONDITIONS

- **Has Government Classification**

- **Has No Government Classification**

- **User Has Government Clearance**

- **User Is ITAR Licensed**

User Is IP Licensed

CATEGORY

Intellectual property (IP)

DESCRIPTION

Checks whether the user being evaluated is listed on an IP license attached to the workspace object.

CONDITION EVALUATION

true
- If set to **true**, the condition returns **true** if the user being evaluated is cited in any valid (not expired) IP license attached to the workspace object being evaluated either directly or as a member of a group.

- If set to **false**, the condition returns **true** if the user being evaluated is *not* cited in any valid (not expired) IP license attached to the workplace object being evaluated either directly or as a member of a group.

false
- If set to **true**, the condition returns **false** if the user being evaluated is *not* listed in any valid (not expired) IP license attached to the workspace object being evaluated either individually or as a member of a group.

- If set to **false**, the condition returns **false** if the user being evaluated is listed in any valid (not expired) IP license attached to the workspace object being evaluated either individually or as a member of a group.

INPUT ARGUMENTS

true or **false**

RELATED RULE CONDITIONS

- **User Has IP Clearance**

- **Has IP Classification**

- **Has No IP Classification**

User Is ITAR Licensed

CATEGORY

International Traffic in Arms Regulations (ITAR)

DESCRIPTION

Checks whether the user currently logged on is cited in a valid (not expired) ITAR license attached to the workspace object either directly or by membership in a cited organization (group).

CONDITION EVALUATION

true
- If the input argument is set to **true**, the condition evaluates to **true** if the user is cited in any valid (not expired) ITAR license attached to the workspace object being evaluated either directly or by membership in a cited organization (group).

- If the input argument is set to **false**, the condition evaluates to **true** if the user is not cited in any valid (not expired) ITAR license attached to the workspace object being evaluated either directly or by membership in a cited organization (group).

false
- If the input argument is set to **true**, the condition evaluates to **false** if the user is not cited in any valid (not expired) ITAR license attached to the workspace object being evaluated either directly or by membership in a cited organization (group).

- If the input argument is set to **false**, the condition evaluates **false** if the user is cited in any valid (not expired) ITAR license attached to the workspace object being evaluated either directly or by membership in a cited organization (group).

INPUT ARGUMENTS

true or **false**

RELATED RULE CONDITIONS

- **Has Government Classification**

- **Has No Government Classification**

- **User Has Government Clearance**

- **User Is Excluded**

User-ITAR Lic Has Citizenship

CATEGORY

Licenses

DESCRIPTION

Checks whether the user's citizenship matches the passed-in value and then checks if the user's citizenship is listed on any of the ITAR licenses attached to the workspace object being evaluated.

CONDITION EVALUATION

true	This condition evaluates to **true** if the user's citizenship matches the input citizenship and that citizenship is listed on any nonexpired ITAR license attached to the workspace object.
false	In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

citizenship	Two-character ISO 3166 codes identifying a country.
	This condition accepts negation using a minus (**–**) prefix. For example, **–IR** means that the user cannot have an IR citizenship.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

• Any workspace object.

RELATED RULE CONDITIONS

• **User-ADA Lic Has Citizenship**

• **User-Exclude Lic Has Citizenship**

• **User-IP Lic Has Citizenship**

User Declared Geography

CATEGORY

International Traffic in Arms Regulations (ITAR)

DESCRIPTION

Checks whether the given geography matches the geography the user declared when logging onto the system.

CONDITION EVALUATION

true	This condition evaluates to **true** if the given geography matches the geography the user declared when logging onto the system.
false	In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

country-code	Two-character ISO 3166 country codes identifying the geography of the user.
	This condition accepts multiple geography arguments, which must be comma separated as shown in the Example section.
	Also, this condition accepts negation using a minus (**-**) prefix. For example, **-US** indicates any user at a site outside the U.S.
	When negating multiple geography arguments, you must enclose the geography arguments in parentheses. For example, **-(US,FR)** indicates the user is not authorized at a site in the U.S. and France.

EXAMPLE

- To specify the geography of a user at a site in the U.S., enter **US**.

- To specify the geography of a user at a site *outside* the U.S., enter **-US**.

- To specify the geography of a user at a site in multiple countries (such as U.S., France, Japan, and India), enter **US,FR,JP,IN**.

- To specify the geography of a user at a site *outside* multiple countries (such as U.S., France, Japan, and India), enter **-(US,FR,JP,IN)**.

RELATED RULE CONDITIONS

- **User Geography**

- **Site Geography**

User Geography

CATEGORY

International Traffic in Arms Regulations (ITAR)

DESCRIPTION

Checks whether the given geography matches the geography of the user being evaluated.

CONDITION EVALUATION

true	This condition evaluates to **true** if the given geography matches the geography of the user being evaluated.
false	In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

country-code	Two-character ISO 3166 country codes identifying the geography of the user.
	This condition accepts multiple geography arguments, which must be comma separated as shown in the Example section.
	Also, this condition accepts negation using a minus (**-**) prefix. For example, **-US** indicates any user at a site outside the U.S.
	When negating multiple geography arguments, you must enclose the geography arguments in parentheses. For example, **-(US,FR)** indicates the user is not authorized at a site in the U.S. and France.

EXAMPLE

- To specify the geography of a user at a site in the U.S., enter **US**.

- To specify the geography of a user at a site *outside* the U.S., enter **-US**.

- To specify the geography of a user at a site in multiple countries (such as U.S., France, Japan, and India), enter **US,FR,JP,IN**.

- To specify the geography of a user at a site *outside* multiple countries (such as U.S., France, Japan, and India), enter **-(US,FR,JP,IN)**.

RELATED RULE CONDITIONS

- **Site Geography**

User Nationality

CATEGORY

International Traffic in Arms Regulations (ITAR)

DESCRIPTION

Checks whether the given nationality matches the nationality of the user being evaluated.

CONDITION EVALUATION

true   This condition evaluates to **true** if the given nationality matches the nationality of the user being evaluated.

false   In all other cases, the condition evaluates to **false**.

INPUT ARGUMENTS

nationality   Two-character ISO 3166 country codes identifying the nationality of the user.

This condition accepts multiple nationality arguments, which must be comma separated as shown in the Example section.

Also, this condition accepts negation using a minus (**-**) prefix. For example, **-US** indicates any user at a site outside the U.S.

When negating multiple nationality arguments, you must enclose the nationality arguments in parentheses. For example, **-(US,FR)** indicates the user is not authorized at a site in the U.S. and France.

BUSINESS OBJECT SCOPE

This condition can be used to control access to classified data.

EXAMPLE

- To specify the nationality of a user at a site in the U.S., enter **US**.

- To specify the nationality of a user at a site *outside* the U.S., enter **-US**.

- To specify the nationality of a user at a site in multiple countries (such as U.S., France, Japan, and India), enter **US,FR,JP,IN**.

- To specify the nationality of a user at a site *outside* multiple countries (such as U.S., France, Japan, and India), enter **-(US,FR,JP,IN)**.

RELATED RULE CONDITIONS

- **Group Nationality**

User Not In Attach ADA Lic Ctg

CATEGORY

Intellectual Property (IP)

DESCRIPTION

Supports the negative rule tree condition for the existing **User In Attach ADA Lic of Ctgry** rule tree condition.

Checks the following:

- Whether the evaluation object is a workspace object (**Workspace Object**) or one of its subtypes.

- The workspace object has ADA licenses attached that do *not*:

  - Match the license category of the input category.

  - List the current session user on the license.

CONDITION EVALUATION

true	Evaluates to **true** if:

Evaluates to **true** if:

- The evaluation object is a workspace object or one of its subtypes.

- The workspace object has ADA licenses attached that do *not*:

  - Match the license category of the input category.

  - List of the current session user on the license.

false Evaluates to **false** if:

- The evaluation object is *not* a workspace object.

- The workspace object has no ADA licenses attached.

- If ADA licenses are attached, none of them list both the user and match the category.

INPUT ARGUMENTS

license_category   A string identifying the name of the license category.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Any workspace object.
  If the evaluated object is a workspace object, the condition returns **true**.

GOOD RULE PRACTICES

Access control by licenses can be configured based on the license type to vary access at a high level or based on the license name to vary the access at a granular level. Categories offer a way to control access by licenses in between the high and granular levels. They provide a way to have different subtypes of licenses under each type and configure access based on each category.

To learn more about categories, see Security Administration.

RELATED RULE CONDITIONS

- **User In Attach ADA Lic of Ctgry**

User Not In Attach Excl Lic Ctg

CATEGORY

Intellectual Property (IP)

DESCRIPTION

Supports the negative rule tree condition for the existing **User In Attach Excl Lic of Ctgry** rule tree condition.

Checks the following:

- Whether the evaluation object is a workspace object (**WorkspaceObject**) or one of its subtypes.

- The workspace object has exclude licenses attached that does *not*:

  - Match the license category of the input category.

  - List the current session user on the license.

CONDITION EVALUATION

true	Evaluates to **true** if:

  - The evaluation object is a workspace object or one of its subtypes.

  - The workspace object has exclude licenses attached that do *not*:

    - Match the license category of the input category.

    - List the current session user on the license.

false	Evaluates to **false** if:

  - The evaluation object is not a workspace object.

  - The workspace object has no exclude licenses attached.

  - If exclude licenses are attached, none of them list both the user and match the category.

INPUT ARGUMENTS

license_category   A string identifying the name of the license category.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Any workspace object.
  If the evaluated object is not a workspace object, the condition returns **false**.

GOOD RULE PRACTICES

Access control by licenses can be configured based on the license type to vary access at a high level or based on the license name to vary the access at a granular level. Categories offer a way to control access by licenses in between the high and granular levels. They provide a way to have different subtypes of licenses under each type and configure access based on each category.

To learn more about categories, see Security Administration.

RELATED RULE CONDITIONS

- **User Not In Attach ADA Lic of Ctgry**

- **User Not In Attach IP Lic of Ctgry**

User Not In Attach IP Lic Ctg

CATEGORY

Intellectual Property (IP)

DESCRIPTION

Supports the negative rule tree condition for the existing **User In Attach IP Lic of Ctgry** rule tree condition.

Checks the following:

- Whether the evaluation object is a workspace object (**WorkspaceObject**) or one of its subtypes.

- The workspace object has IP licenses attached that:

    - Do not match the license category of the category input.

    - Do not list the current session user on the license.

CONDITION EVALUATION

true   Evaluates to **true** if:

- The evaluation object is a workspace object or one of its subtypes.

- The workspace object has any IP licenses attached that:

    - Does not match the license category of the input category.

    - Does not list the current session user on the license.

false   Evaluates to **false** if:

- The evaluation object is a workspace object or one of its subtypes.

- The workspace object has IP licenses attached.

- If ITAR licenses are attached, both of them list both the user and match the category.

INPUT ARGUMENTS

license_category   A string identifying the name of the license category.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

- Any workspace object.
  If the evaluated object is a workspace object, the condition returns **true**.

GOOD RULE PRACTICES

Access control by licenses can be configured based on the license type to vary access at a high level or based on the license name to vary the access at a granular level. Categories offer a way to control access by licenses in between the high and granular levels. They provide a way to have different subtypes of licenses under each type and configure access based on each category.

To learn more about license categories, see Security Administration.

RELATED RULE CONDITIONS

- **User Not In Attach ADA Lic of Ctgry**

- **User Not In Attach Excl Lic of Ctgry**

User Not In Attach ITAR Lic Ctg

CATEGORY

International Traffic in Arms Regulations (ITAR)

DESCRIPTION

Supports the negative rule tree condition for the existing **User In Attach ITAR Lic of Ctgry** rule tree condition.

Checks the following:

- Whether the evaluation object is a workspace object (**Workspace Object**) or one of its subtypes.

- The workspace object has ITAR licenses attached that do *not*:

    - Match the license category of the input category.

    - List the current session user on the license.

CONDITION EVALUATION

true	Evaluates to **true** if:

- The evaluation object is a workspace object or one of its subtypes.

- The workspace object has ITAR licenses attached that do *not*:

    - Match the license category of the input category.

    - List the current session user on the license.

false	Evaluates to **false** if:

- The evaluation object is not a workspace object.

- The workspace object has no ITAR licenses attached.

- If ITAR licenses are attached, none of them list both the user and match the category.

INPUT ARGUMENTS

license_category   A string identifying the name of the license category.

BUSINESS OBJECT SCOPE

This condition can be used to control access to:

• Any workspace object.
  If the evaluated object is not a workspace object, the condition returns **false**.

GOOD RULE PRACTICES

Access control by licenses can be configured based on the license type to vary access at a high level or based on the license name to vary the access at a granular level. Categories offer a way to control access by licenses in between the high and granular levels. They provide a way to have different subtypes of licenses under each type and configure access based on each category.

To learn more about license categories, see Security Administration.

RELATED RULE CONDITIONS

• **User Not In Attach ADA Lic of Ctgry**

• **User Not In Attach IP Lic of Ctgry**

• **User Not In Attach Excl Lic of Ctgry**

User TTC Expired

CATEGORY

International Traffic in Arms Regulations (ITAR)

DESCRIPTION

Checks whether the current date is later than the technology transfer certification (TTC) date on the **User** object.

CONDITION EVALUATION

true	• If the current date is later than the TTC value on the **User** object, the condition evaluates to **true**.
false	• If the current date is earlier than the TTC value on the **User** object, the condition evaluates to **false**.

> Note:
>
> If the TTC value on the **User** object is not entered, the condition evaluates to **true**.

INPUT ARGUMENTS

Current date	Specifies today's date.
Technology Transfer Certification (TTC) date	Specifies the technology transfer certification date, which is the date when the user's qualification for viewing exporting data marked as government classified lapses.

BUSINESS OBJECT SCOPE

This condition can be used to control access to classified data.

RELATED RULE CONDITIONS

• **Has Government Classification**

• **Has No Government Classification**

• **User Has Government Clearance**

• **User Is Excluded**

Best practices for rules

- **Understand your organization's business rules.**

  A thorough understanding of your organization's business rules enables you to model access rules that support your business processes and are transparent to users. When modeled correctly, Access Manager rules grant users the privileges required to perform the tasks associated with their jobs while denying them access to data that is released or out of the scope of their functional role.

- **Document the business rules and the rule tree developed to meet them.**

  Every rule in the rule tree and the named ACLs associated with the rules are included for a purpose. For maintenance purposes, Siemens Digital Industries Software strongly recommends that you document the purpose of the rules, how they are populated, and why they have been populated. Future versions of Teamcenter add new rules and accessors. Merging new rules and accessors is a manual process, which is simplified if you have thoroughly documented the Access Manager rule tree.

- **Export the rule tree before and after making changes.**

  When new rules do not work as expected, you must be able to restore an earlier, working version of the rule tree. A backup copy is essential to restoring rules back to their original state.

- **Add new rules for working data in the Working data branch of the tree.**

  The proper location to add new rules for working data is under the **Working** data branch in the rule tree. This helps you customize your rule tree and identify working data.

  ```
  Has Class(POM_application_object) -> Working
  ```

- **Whenever possible, leave privileges unset.**

  Leaving privileges unset in ACLs allows rules to accomplish focused objectives, and it also allows objects and accessors to filter through rules that do not apply to them.

- **Populate access control lists (ACLs) sparingly.**

  Explicitly grant privileges, and only deny privileges when you must block users from access that would otherwise be implicitly granted.

- **Use the Has Attribute condition to create custom rules based on any attribute of an object of a given class.**

  For example:

  ```
  WorkspaceObject:object_name=*x
  PublicationRecord:security=suppliers
  ```

  The class and attribute names are not case sensitive. The attribute type can be **string**, **double**, **integer**, **logical**, or **reference**.
  This rule supports custom attributes.

- **When using ip_classification or gov_classification attributes, use the Has IP Classification rule or the Has Government Classification rule, instead of the Has Attribute rule.**

  For example, use the **Has IP Classification** rule:

  ```
  Has IP Classification ( secret ) -> Secret ACL
  ```

  Instead of using the **Has Attribute** rule:

  ```
  Has Attribute ( WorkspaceObject:ip_classification=secret ) -> Secret
  ACL
  ```

- **Use the Has Property condition to create custom rules based on the value of compound properties.**

  For example:

  ```
  Item:my_custom_prop=my_custom_prop_value
  ```

  In this example, **Item** is the type name and **my_custom_prop** is the compound property.

- **Set security precedence.**

  You can embed type-level security rules under project-level security rules to give the type-level security rules higher precedence than the project-level security rules. For example, the project administrator can add a subbranch under the **Has Class (Form)** rule entry to control access to certain form types that contain sensitive data. The rule for the form type is written as follows:

  ```
  Has Class(Form)
      Has Type(Finance) -> finance_acl
  ```

  If your site requires that project-level security rules take precedence over type-level security rules, you must embed project-level security rules under the type-level security rules. However, Siemens Digital Industries Software does not recommend this practice.

- **Define relevant ACL names.**

  ACL names are displayed in the rule tree and in dialog boxes throughout the Teamcenter interface. You can significantly enhance overall usability by defining these names carefully. For example, when creating an ACL for working data, name it according to the data type (for example, item, item revision, or **UGMASTER**) rather than a role name or some other description.

  > Note:
  >
  > ACLs can be referenced in more than one rule.

- **Use discretion in applying the Bypass ACL.**

The **Bypass** ACL grants all privileges to system administrators who have selected the user **Bypass** setting. Use discretion in applying this ACL.

• **Do not create GRM relations**

Do not create Generic Relationship Management (GRM) relationships between Teamcenter business objects, such as BOM View, and Access Manager objects, such as AM Tree, Named ACL, and AM_ACE. Creating such relationships can result in unpredictable behavior with Access Manager during run time.

Cautions for using rule trees

• **Do not modify access control lists (ACLs) referenced by rules on the System Objects branch.**

Adding new rules, deleting rules, or in any way modifying existing rules on the **Systems Objects** branch of the rule tree may result in unpredictable behavior or loss of data. Modifying the **Systems Objects** branch of the rule tree is not supported unless specifically advised to do so by Siemens Digital Industries Software.

• **Do not modify the upper area of the rule tree.**

Deleting or changing the order of the branches in this area of the rule tree may result in unpredictable behavior or loss of data.

• **Do not use a text editor to modify rule tree files.**

Rule tree files are simple ASCII files and conform to a particular format. You can read rule tree files using any text editor; however, modifying them with a text editor can easily corrupt the file.

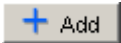• **Do not use the infodba account to change object ACLs.**

It is assumed that objects owned by **infodba** are seed parts or other special-case objects.

Add an Access Manager rule

1. Select the parent tree rule to which the new node will be added.

2. Set the **Condition**, **Value**, and **ACL Name** for the new rule.

    > Note:
    >
    > ACLs can be referenced in more than one rule.

3. Click the **Add** ![+ Add] button located below the ACL table.

4. Click the **Save** button ![save icon] in the toolbar.

This creates the new rule and adds it to the selected parent in the rule tree. An asterisk appears next to the Access Manager name indicating that the application has been modified.

Modify an Access Manager rule

1. Select the rule you want to modify.

2. Modify the condition or value in the rule pane.

3. To attach an ACL to the rule, select an ACL from the **ACL Name** list.

4. Click the **Modify** ⚷ button located below the ACL table.

5. Click the **Save** 💾 button in the toolbar.

> Note:
>
> When you make changes to a rule, the changes are not saved until you choose **File**→**Save** or click the **Save** 💾 button on the toolbar.

Delete an Access Manager rule

1. Select the rule you want to delete.

2. Click the **Delete** button ❌ Delete located below the ACL table.

3. Click the **Save** button 💾 in the Access Manager toolbar.

> Note:
>
> Deleting a rule does not delete its corresponding ACLs. To remove ACLs from the rule tree, they must be explicitly **deleted**.

Reposition an Access Manager rule in the rule tree

1. Select the rule that you want to reposition.

2. After selecting the rule, you can:

    *   Click **Move Up** ⬆ in the toolbar to move the rule up one level in the rule tree.

- Click **Move Down** ⬇ in the toolbar to move the rule down one level in the rule tree.

3. Click **Save** 🖫.

Managing your administrative data

There are different types of administration data, for example, Access Manager rules and Organization data. At times, it is necessary to manage administrative data by moving data between your development and production environments. Because administration data is locally owned, moving this data between sites is handled differently from shared data.

To ensure proper operation, both sites should share the same Teamcenter version to ensure proper operation. However, if both sides have the same data model for the data being exchanged, the exchange can occur with different versions of Teamcenter and still operate properly.

You can use Teamcenter Environment Manager (TEM) to manage your administration data at multiple sites. For example, you can export and import administration data using panels in TEM that are accessed through the **Manage Administration Data** option in the **Feature Maintenance** panel. Using TEM, you can select the specific instances of administration data by category, class, and specific attribute/value criteria.

You can do the following:

- Generate and view a report containing Access Manager administration data using the **generate_admin_data_report** utility.

- Generate and view a report comparing administration data at two sites using the **generate_admin_data_compare_report** utility.

- Export Access Manager named ACLs and privileges using the **admin_data_export** utility.

- Import Access Manager administration data using the **admin_data_import** utility.

4. Creating and managing access control lists (ACLs)

Types of access control lists (ACLs)

There are three types of ACLs:
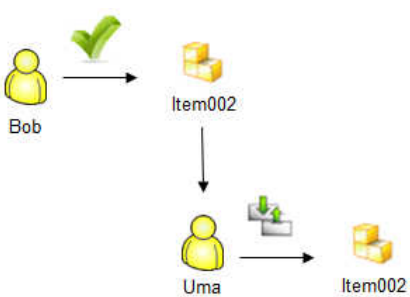
- Rule tree ACL
  These ACLs control access to general data creation. They are managed through Access Manager.

- Workflow ACL
  These ACLs control access to data that is in process at a particular release level. They provide a subset of Access Manager functionality that can be accessed from Workflow Designer.

- Project ACL
  These ACLs control access to project data. They provide a subset of Access Manager functionality that can be accessed from Project.

Access privileges

Symbol	Privilege	Description
	Create	Controls the creation of objects.
		Note: There are best practices for ACLs to consider when creating an ACL with the **Create** privilege.
	Read	Controls the privilege to open and view an object.
	Write	Controls the privilege to check the object in/out of the database and modify it.
	Delete	Controls the privilege to delete the object from the database.
	Change	Controls the privilege to modify object protections that override the rules-based protection for the object. You must have change privileges to apply object-based protection (object ACLs).

Symbol	Privilege	Description
	Promote	Controls the privilege to move a task forward in a workflow process.
	Demote	Controls the privilege to move a task backward in a workflow process.
	Copy	Controls the privilege to copy an object as a new object.

> **Note:**
>
> It still allows copy and paste of the object as a reference, with no new object created.

Symbol	Privilege	Description
	Change ownership	Controls the privilege required to grant, change, or restrict ownership rights to an object.

> **Note:**
>
> **Write** access is required.

Symbol	Privilege	Description
	Publish	Controls the publish privilege to users or groups.
	Subscribe	Controls the privilege to subscribe to an event on a specified workspace object.
	Export	Controls the privilege to export objects from the database.
	Import	Controls the privilege to import objects in to the database.
	Transfer out	Controls the privilege to transfer ownership of objects when they are exported from the database.
	Transfer in	Controls the privilege to assign ownership of objects when they are imported in to the database.
	Write Classification ICO	Controls the privilege to write Classification objects (ICOs).
	Assign to project	Controls the privilege to assign an object to a project. This applies to users who are not designated as privileged project team members.

Symbol	Privilege	Description
		> **Note:** The validation of the **Assign to project** privilege in conjunction with privileged project membership is evaluated based on the value of the **TC_project_validate_conditions** preference.
	Remove from project	Controls the privilege to remove an object from a project. This applies to users who are not designated as privileged project team members. > **Note:** The validation of the **Assign to project** privilege in conjunction with privileged project membership is evaluated based on the value of the **TC_project_validate_conditions** preference.
	Remote checkout	Controls the privilege to remotely check out an object.
	Unmanage	Enables users to circumvent the blocking implemented using the **TC_session_clearance** preference.
	IP Admin	Enables users to add users to manage IP licenses.
	ITAR Admin	Enables users to add administrative users to manage ITAR licenses.
	CICO	Grants a user the ability to override the checkout of an object by another user. It lets the user with the override privilege check in, transfer, or cancel the checkout of the object. > **Example:** If Bob checks out an object (**item2**) and forgets to check it back in before leaving on vacation, the **CICO** privilege can be granted to the project manager, Uma, so she can check **item2** back in and the project can proceed.

Symbol	Privilege	Description
	Translation	Controls the privilege to add translated text using the **Localization** button.
	View/Markup	Controls the privilege to view and create markups.
	Batch Print	Controls the privilege to print multiple objects.
	Digitally Sign	Controls the privilege to digitally sign a document. The Commercial Off-The-Shelf (COTS) Digital Sign Dataset ACL rule grants owning user and owning group digital sign privileges for the dataset object. World users do not have digital sign privileges.
	Void Digital Signature	Controls the privilege to revoke or cancel an existing PKI digital signature for a business object. World users do not have void digital signature privileges.
	Administer ADA Licenses	Controls the privilege to create, modify, or delete ADA licenses for users in the ADA License application.
	IP Classifier	Controls the privilege to classify intellectual property (IP) information.
	ITAR Classifier	Controls the privilege to classify international traffic in arms (ITAR) information.
	Remove Content	Allows a user of Smart Discovery and 4th Generation Design on Rich Client (4GD) to remove content from a collaborative design (CD), for example, to remove an existing design element.
	Add Content	Allows a user of Smart Discovery and 4th Generation Design on Rich Client (4GD) to add content to a CD, for example, to create a new design element.

Symbol	Privilege	Description
	Effectivity	Allows a user to modify effectivity on released objects.
	Manage Variability	Allows a user to add or remove the association between a product structure and a configurator context.
	PDF Control	Allows a user to add a system stamp and watermarks to an existing PDF using workflow.

Accessor precedence

An *accessor* is a user or group of users who share certain traits, such as membership in the group that owns the object or membership in the project team. The following list presents the predefined accessors delivered with Teamcenter in order of precedence, from most restrictive to least restrictive. The more restrictive the accessor, the higher precedence it has over other accessors.

> Note:
>
> - When two accessors with different precedences are added to a named ACL configuration, the highest precedence accessor is automatically moved to the top in the ACL table.
>
> - When two accessors with the same precedence are added to a named ACL configuration, they stay in the order they are added.
>
> - The **Role in Group**, **Role in Owning Group**, **Role in Project**, and **Role in Project of Object** accessors work on the superset of roles the user possesses in the relevant group or project, rather than on the session current role.
>
> - When the **TC_current_role** preference is set, it affects the evaluation of the **Role in Owning Group**, **Role in Group**, and **Role** accessors. It enforces object access based on the user's current role in the current group.
>
> - When the **AM_PROJECT_MODE** preference is set, it affects the evaluation of the **Role in Project** and **Role in Project of Object** accessors.

Most restrictive - Highest precedence

Approver (RIG)
Approver (Role)
Approver (Group)
Approver
Task Owner
Task Owning Group
Responsible Party
Owning User
User Excluded
User ITAR Licensed
User ITAR Unlicensed
User Under Government Clearance
User Has Government Clearance
User Over Government Clearance
User IP Licensed
User IP Unlicensed
User Under IP Clearance
User Has IP Clearance
User Over IP Clearance
User in License
User not in License
User
Group Administrator
Role in Owning Group
Role in Projects of Object
Role in Group
Role in Project
Role
Owning Group
System Administrator Group
Group
Current Project Team
Current Project Teams
Project Team
Groups with Security
Project Teams
RoleInSchedule
Public Schedule
Read Schedule
Regular Project Member
Project Administrator
Project Team Administrator
Privileged Project Member
World
Site
Remote Site
Resource graph viewers

Least restrictive - Lowest precedence

Accessor types by category

The following table lists the accessor types by category.

Accessor type	Accessor (input argument)	Description
General		
Owning User	Not applicable	Evaluates any **POM_application_object**.
		Evaluates to true if the current logged-on user matches the user listed on the **owning_user** attribute of the object being evaluated.
		Example:
		```ObjecA.owning_user=User1```
		If **User1** logs on, this accessor type evaluates to true.
		If **User2** logs on, this accessor type evaluates to false.
Owning Group	The group of the user who first created the object	Evaluates to true if the current logged-on user's group membership is the group listed on the **owning_group** attribute of the object being evaluated. The **owning_group** attribute is always set to the group of the user who first created the object.
		Additional privileges (for example, write) may be granted to the owning group, because it is common for users to share data with other members of their group.
		Note: By default, members of a subgroup receive the same access privileges set on workspace objects as their parent group who owns the object (the owning group). To change the privilege inheritance, use the **TC_allow_group_hierarchy_traversal** preference.
		Example:
		```ObjecA.owning_group=Group1```
		If **Group1** logs on, this accessor type evaluates to true.
		If **Group2** logs on, this accessor type evaluates to false.
Group	Any group named in the Organization application	Evaluates to true if the current logged-on user's group membership matches the current logged-on user's group.
		Example:
		```Accessor Type=Group and the Accessor=Group1```
		If **User1** logs on as a member of **Group1**, this accessor type evaluates to true.
		If **User1** logs on as a member of **Group2**, this accessor type evaluates to false.
Groups with Security	A user whose group has the given security value, either **Internal** or **External**	Evaluates to true if the current logged-on user has the given security value, either **Internal** or **External**. This value is used to distinguish between groups in the parent company (internal) and suppliers (external).

Accessor type	Accessor (input argument)	Description
		Example:
		``` Accessor Type=Groups with Security and the Accessor=Internal ```
		If **Group1** user logs on as **Internal** (for example, company employee), this accessor type evaluates to true.
		If **Group2** user logs on as a member of **External** (for example, supplier), this accessor type evaluates to false.
Role	Any role named in the Organization application	Evaluates to true if the current logged-on user's role membership matches the current logged-on user's role.
		Example:
		``` Accessor Type=Role and the Accessor=Role1 ```
		If **User1** logs on as **Role1**, this accessor type evaluates to true.
		If **User1** logs on as **Role2**, this accessor type evaluates to false.
Role in Group	A specific role	Evaluates to true if the current logged-on user performs the same skills and/or responsibilities as other users on the same project.
		Example:
		``` Accessor Type=Role in Group and the Accessor=TranslatorFrench ```
		If **User1** logs on as **TranslatorFrench**, this accessor type evaluates to true.
		If **User1** logs on as **TranslatorSpanish**, this accessor type evaluates to false.
Role in Owning Group	A specific role	Evaluates to true if the current logged-on user's role grants specific privileges. For example, all designers in the owning group are usually granted write privilege on their development data.
		Example:
		``` Accessor Type=Role in Group and the Accessor=Designer ```
		If **User1** logs on as **Designer**, this accessor type evaluates to true.
		If **User2** logs on as **Consultant**, this accessor type evaluates to false.
System Administrator	A user who is a member of the system administration group	Evaluates to true if the current logged-on user is a member of the system administration group.
		Example:
		``` Accessor Type=System Administrator and the Accessor=SystemAdministrationGroup ```
		If **User1** logs on as belonging to **SystemAdministrationGroup**, this accessor type evaluates to true.

Accessor type	Accessor (input argument)	Description
		If **User1** logs on as belonging to **Group2**, this accessor type evaluates to false.
Group Administrator	A user who has special maintenance privileges for the group	Evaluates to true if the current logged-on user has group administrator privileges. A group administrator is a group member who can add, modify, or remove group members.
		Example:
		``` Accessor Type=Group Administrator and the Accessor=User1 ```
		If **User1** logs on, this accessor type evaluates to true.
		If **User2** logs on, this accessor type evaluates to false.
Site	Any site named in the Organization application	Evaluates to true if the current logged-on site (Teamcenter installation) matches the site listed on the site attribute of the object being evaluated.
		Example:
		``` Accessor Type=Site and the Accessor=Site1 ```
		If **User1** logs on as being on **Site1**, this accessor type evaluates to true.
		If **User1** logs on as being on **Site2**, this accessor type evaluates to false.
Remote Site	Any remote site	Evaluates to true if the current logged-on remote site (Teamcenter installation) matches the remote site listed on the remote site attribute of the object being evaluated.
		Example:
		``` Accessor Type=Remote Site and the Accessor=RemoteSite1 ```
		If **User1** logs on as being on **RemoteSite1**, this accessor type evaluates to true.
		If **User1** logs on as being on **RemoteSite2**, this accessor type evaluates to false.
World	Any user on the system	Evaluates to true, as this represents *all* users.
		Example:
		``` Accessor Type=World and the Accessor=User1 ```
		If **User1** logs on as **World**, this accessor type evaluates to true.
		If **User2** logs on as **World**, this accessor type evaluates to true.
User	Any user named in the Organization application	Evaluates to true if the current logged-on user matches the user listed on the user attribute of the object being evaluated.
User In License	A specific user	Evaluates to true if the current logged-on user is listed on the license either through the user or group value.
		The term ADA license refers to any ITAR, IP, or exclude license.
		Example:

Accessor type	Accessor (input argument)	Description
		`Accessor Type=User and the Accessor=User1` If **User1** logs on as **User,** this accessor type evaluates to true. User is not listed in the ADA_License object being evaluated. If **User2** logs on as **User,** this accessor type evaluates to false.
Workflow		

Approver (RIG)	Any role that is designed as an approver in the workflow process.	Evaluates to true if the current logged-on user's role matches the user who is a signoff team member in the workflow process for the group.
		
		> Note:
		> This accessor must only be used in a workflow ACL and match the signoff **RIG** requirements for the release level associated with the workflow ACL.
		
		Example:
		
		`Accessor Type=Approver (RIG)`
		`and the Accessor=Override Approver`
		`in Validation Administration`
		
		If **User1** logs on as **Override Approver in Validation**, this accessor type evaluates to true.
		
		If **User2** logs on as **Designer in Engineering**, this accessor type evaluates to false.

Approver (Role)	Any user designed as an approver in the workflow process.	Evaluates to true if the current logged-on user's role matches the user who is a signoff team member in the workflow process for the group.
		
		> Note:
		> This accessor must only be used in a workflow ACL.
		
		Example:
		
		`Accessor Type=Approver (Role)`
		`and the Accessor=Approver`
		
		If **User1** logs on with **Approver** role, this accessor type evaluates to true.
		
		If **User2** logs on with **Designer** role, this accessor type evaluates to false.

Approver (Group)	Any group that is designed as an approver in the workflow process.	Evaluates to true if the current logged-on user's role matches the user who is a signoff team member in the workflow process for the group. > Note: > This accessor must only be used in a workflow ACL. **Example:**

Accessor type	Accessor (input argument)	Description
| | | ```
Accessor Type=Approver (Group)
and the Accessor=Engineering
``` |
		If **User1** logs on as a member of the **Engineering** group, this accessor type evaluates to true.
		If **User2** logs on as a member of the **Simulation** group, this accessor type evaluates to false.
Approver	Any user designed as an approver in the workflow process.	Evaluates to true if the current logged-on user, who is a signoff team member, has approver privileges.
		> **Note:** > This accessor must only be used in a workflow ACL.
		Example:
| | | ```
Accessor Type=Approver and the Accessor=User1
``` |
		If **User1** logs on, this accessor type evaluates to true.
		If **User2** logs on, this accessor type evaluates to false.
Task Owner	A user who is granted privileges for the task's target data.	Evaluates to true if the current logged-on user has task owner privileges for the task's target data.
		> **Note:** > This accessor must only be used in a workflow ACL.
		Example:
| | | ```
Accessor Type=Task Owner
and the Accessor=User1
``` |
		If **User1** logs on, this accessor type evaluates to true.
		If **User2** logs on, this accessor type evaluates to false.
Task Owning Group	A group that is granted privileges for the task's target data.	Evaluates to true if the current logged-on user is a member of the task owning group.
		> **Note:** > This accessor must only be used in a workflow ACL.
		Example:
| | | ```
Accessor Type=Task Owning Group
and the Accessor=OwningGroup
``` |
| | | If **User1** logs on as a member of **OwningGroup**, this accessor type evaluates to true. |

Accessor type	Accessor (input argument)	Description
		If **User2** logs on as a member of **Engineering**, this accessor type evaluates to false.
Responsible Party	A user assigned as responsible for performing a particular task	Evaluates to true if the current logged-on user is the person responsible for performing a particular task.

> Note:
>
> This accessor must only be used in a workflow ACL.

Example:

```
Accessor Type=Responsible Party
and the Accessor=User1
```

If **User1** logs on, this accessor type evaluates to true.

If **User2** logs on, this accessor type evaluates to false.

Project		
Project Team	Users in a project team to which the object is assigned	Evaluates to true if the current logged-on user is an active group member in a project team to which the object is assigned.

Example:

```
Accessor Type=Project Team and the Accessor=User1
```

If **User1** logs on, this accessor type evaluates to true.

If **User2** logs on, this accessor type evaluates to false.

> Note:
>
> This does not apply to project team members who are inactive group members.

Project Teams		Team members (active group members) in any active project for the object.

> Note:
>
> This does not apply to project team members who are inactive group members.

Current Project Team		Users who are members of a particular current project team. Applicable only when the project is set as the current project of the team members and if the current project is active.
Current Project Teams		Users who are members of current project teams. Applicable only when the object is in the current project of the team members, and the current project is active.
Regular Project Member		Evaluates to true if the logged-on user is a regular team member of the **TC_project** object on which access is being evaluated.

Accessor type	Accessor (input argument)	Description
Administrator Project Member		Evaluates to true if the logged-on user is the administrator of the **TC_project** object on which access is being evaluated.
Team Admin Project Member		Evaluates to true if the logged-on user is the team administrator of the **TC_project** object on which access is being evaluated.
Privileged Project Member		Evaluates to true if the logged-on user is the privileged team member of the **TC_project** object on which access is being evaluated.
Role in Projects of Object		Users who have a specific role in one of the projects of the object. This accessor is affected by the values set in the **AM_PROJECT_MODE** preference. It is effective only when the user is logged-on with the specified role in the current project, and the current project is one of the projects assigned to the defined object.
Role in Project		Project members with a specific role in a specific project. This is affected by the values set in the **AM_PROJECT_MODE** preference.
Scheduler		
Public Schedule		Access to all users for schedules that are templates or made public. This accessor applies to the Schedule Manager application.
RoleInAnySchedule		Membership privileges of the logged-on user across all schedules in the system. Member privileges (accessor IDs) can be **COORDINATOR**, **PARTICIPANT**, or **OBSERVER**. This accessor applies to the Schedule Manager application.
ADA		
User In License	Not applicable.	This accessor type controls access to a workspace object.
		This accessor type evaluates to true if the current logged-on user is listed on any ADA license attached to the object being evaluated.
		During evaluation, the accessor type looks at the attached licenses. The accessor type will evaluate to true if the current logged-on user is listed on any license, or the user is a member of a group that is listed on the license.
		The term ADA license refers to any ITAR, IP, or exclude license.

> **Note:**
>
> If there are no licenses attached to the object being evaluated, this accessor type evaluates to false.

Example:

Use case scenario:

License1 lists **User1**, **Group2**.

User2 is a member of **Group2**.

ObjectA attaches License1.

Evaluation results:

- **User1** evaluates to true because **User1** is listed on License1.

- **User2** evaluates to true because **User2** is a member of **Group2**, which is listed on License1.

Accessor type	Accessor (input argument)	Description
		• **User3** evaluates to false because **User3** is not listed on License1 nor is **User3** a member of a group listed on License1.
User Not In License	Not applicable	Evaluates to true if the current logged-on user is *not* listed on the license either through the user or group value.
		The term ADA license refers to any ITAR, IP, or exclude license.
		Example:
		``` Accessor Type=User Not In License and the Accessor=LicenseObject ```
		If **User1** logs on and is not listed in **LicenseObject**, this accessor type evaluates to true.
		If **User2** logs on and it listed in **LicenseObject**, this accessor type evaluates to false.
User Excluded	Not applicable.	The user or group is listed in a valid exclude license attached to the workspace object being evaluated.
		The term ADA license refers to any ITAR, IP, or exclude license.
		Example:
		``` Accessor Type=User Excluded and the Accessor=User1 ```
		If **User1** logs on with a valid exclude license, this accessor type evaluates to true.
		If **User2** logs on with no valid exclude license, this accessor type evaluates to false.
ITAR		
User Has Government Clearance		Compares the user's clearance with the object classification and tests whether the user has clearance above, below, or equal to that required to access the object.
User ITAR Licensed	Not applicable.	Evaluates to true if the current logged-on user is cited in a current license associated with the selected object.
		Example:
		``` Accessor Type=User ITAR Licensed and the Accessor=License1 ```
		If **User1** logs on with **License1**, this accessor type evaluates to true.
		If **User2** logs on with no **License1**, this accessor type evaluates to false.
User ITAR Unlicensed		The user is not cited in a current license associated with the selected object.
User Under Government Clearance		The user's clearance is below the level required by the object. This accessor is typically used to revoke access and is only applicable when the government clearance on the user and the government classification on the object come from a common multi-level scheme defined by the **ITAR_level_list_ordering** preference.

Accessor type	Accessor (input argument)	Description
User Over Government Clearance		The user's clearance is over the level required by the object. This accessor is typically used to grant access and is only applicable when the government clearance on the user and the government classification on the object come from a common multilevel scheme defined by the **ITAR_level_list_ordering** preference.
IP		
User Is IP Licensed	Any user cited in a current license associated with the selected object either directly or by membership in a cited group	Evaluates to true if the current logged-on user is cited in a current license associated with the selected object either directly or by membership in a cited organization (group). **Example:** ``` Accessor Type=User IP Licensed and the Accessor=User1 ``` If **User1** logs on as **User**, this accessor type evaluates to true. If **User2** logs on as **User**, this accessor type evaluates to false.
User IP Unlicensed		The user is not cited in a current license associated with the selected object.
User Has IP Clearance		Compares the user's clearance (**secret**, **super-secret**, **top-secret**) with the object classification and tests whether the user has clearance above, below, or equal to that required to access the object.
User Over IP Clearance		The user's clearance is over the level required by the object. This accessor is typically used to grant access and is only applicable when the IP clearance on the user and the IP classification on the object come from a common multi-level scheme defined by the **IP_level_list_ordering** preference.
User Under IP Clearance		The user's clearance is below the level required by the object. This accessor is typically used to revoke access and is only applicable when the IP clearance on the user and the IP classification on the object come from a common multi-level scheme defined by the **IP_level_list_ordering** preference.

Best practices for ACLs

There are three approaches to restrict a user from creating certain business objects.

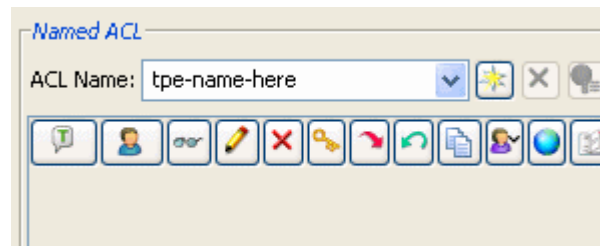| Restrict object creation	Where in Teamcenter
Suppress business objects by organization	Business Modeler IDE display rules
Suppress Teamcenter menus or commands by organization	Teamcenter Command Suppression
Suppress on Access Manager class or type by user, group, or role	**Access Manager using Create ✸ in a named ACL**

If you create a named ACL with the **Create** privilege to grant ✔ or deny ✘ permissions for users to create objects, there are certain business objects where creation is not controllable.

> Note:
>
> Some AM rules do not lend themselves to using **Create**. For example, you would not include **Has Status** because it is impossible for a business object to have status populated at time of creation.

Create an access control list (ACL)

1. In the **Named ACL** section of the Access Manager, enter the ACL name in the **ACL Name** box.

2. Click **Create** ✦ to the right of the **ACL Name** box.

3. Click **Add** ➕ to add a new row to the access control entry (ACE) table.

4. Double-click the cell in the **Type of Accessor** column to select an accessor.

5. Double-click the cell in the **ID of Accessor** column to select an accessor ID.

> Note:
>
> Some accessor types, such as **User**, **Group**, and **Role**, require you to select an accessor ID to define a specific instance of the accessor type. Other accessor types, such as **World** and **Owning Group**, are either singular or are relative to the object being accessed; therefore, no ID is required.

6. Set privileges by double-clicking the cell corresponding to the privilege you want to set, and choose ✔ to grant privileges or choose ✘ to deny privileges.

> Note:
>
> Whenever possible, do not explicitly set privileges. Leaving privileges unset allows rules to accomplish focused objectives by allowing objects and accessors to filter through rules that do not apply to them.

7. (Optional) Click **Localization** 🌐 to display the **Language Translations** dialog box and set localized values for the ACL.

8. Click **Save** 💾.

Modify an access control list (ACL)

1. Select the ACL you want to change from the **ACL** list.

   > Note:
   >
   > You cannot modify the **Accessor Type** or **Accessor ID** values. To change these values, you must delete the entry and add a new entry that reflects the correct accessor type and ID.

2. Modify the privileges.

3. (Optional) Click **Localization** 🌐 to display the **Language Translations** dialog box and set localized values for the ACL.

4. Click **Save** 💾.

Delete an access control list (ACL)

1. Select the ACL you want to delete from the **ACL** list.

2. Click **Delete ACL** ❌ .

3. Click **Save** 💾.

5. Distributing, reverting, and repairing the rule tree

About distributing, reverting, and repairing the rule tree

Importing and exporting the rule tree file enables you to distribute access rules to other Teamcenter sites and also enables you to restore your local rule tree file.

> Note:
>
> Rules, ACLs, accessors, and privileges that support new functionality are introduced with each Teamcenter version. Introducing new rules into your security implementation requires analysis to determine how they should be used.

You can distribute rules to other sites by first exporting the rule tree as an ASCII file and then importing that file at the receiving site.

Before importing a rule tree file, you must ensure schema compatibility. To successfully load a new rule tree from a file, the importing site must have the same types, roles, and groups as those referenced in the rule tree file. If there is any incompatibility, the import operation ends at the first discrepancy and an error message appears.

If you encounter schema compatibility issues, open the rule tree file with a text editor and either print the file or make note of the types, roles, and groups referenced in the file. You can then use the Organization application to define the exact types, roles, and groups at your site.

> Caution:
>
> Siemens Digital Industries Software recommends that you do not modify the rule tree file in a text editor, as this file must conform to a particular format and can be easily corrupted. You can use Access Manager to modify the rule tree after the file is imported.

Reverting the rule tree to a previous version

You can export your access rules before making major changes to the rule tree, which enables you to import the file if the rules need to be restored. Another method of restoring the rule tree is to import the file that is created each time the rule tree is saved.

When you save the rule tree, a file is saved in the **TC_DATA\am** directory. This file is named **tree_***date-time*; it can be used to revert the rule tree to its state at a specific date and time.

Speeding up Solr reindexing after AM rule tree modifications

If you deploy Active Workspace and modify the AM rule tree, a re-indexing of objects in the Solr cache occurs. This slows down saving the rule tree. To save the rule tree faster, set the **TC_SKIP_FINDINGS_AM_IMPACTED_OBJECTS** environment variable to any value, for example, **true** or **ON**. Setting this environment variable causes the system to skip an intermediate step that queues up objects for automatic re-indexing in the accountability table.

> Note:
>
> If you use the **TC_SKIP_FINDINGS_AM_IMPACTED_OBJECTS** environment variable, you must run indexing manually to submit the modifications to the Solr engine directly; this enables the security strings on each object to reflect the rule changes.

Access Manager bypass for administrators

The **AM_BYPASS** environment variable can be used to allow administrators to bypass Access Manager rules.

This enables you to repair the rule tree in the event that rule tree modifications have been made that render you unable to functionally log on to Teamcenter. For example, if a rule tree modification results in rendering you unable to see your **Home** folder when you log on to Teamcenter, you can use the bypass privilege to log on and repair the rule tree.

Setting this environment variable to any value prompts the system to bypass the AM rule tree when logging on.

> Note:
>
> This environment variable should only be used when you cannot log on to Access Manager using your standard administrative logon. It is not intended for general rule tree maintenance.

Export the Access Manager rule tree

Access Manager exports the rule tree in XML file format.

1.   Choose **File→Export**.

2.   Enter a name for the file into which you want to export the AM rule tree data and browse to the directory where you will store the new file.

3.   Click **Export**.

Import the Access Manager rule tree

1. Choose **File→Import**.

2. Locate the XML file to be imported.

3. Click **Import**.

Merge a new system branch

If you have one of the following situations, you must update your Teamcenter environment to accommodate it:

- After upgrading to new version of Teamcenter, you notice there is a new system branch in the default rule tree for that version. However, you cannot add it because the system branch of the rule tree is not modifiable in Access Manager.

- You have many custom stubs and ACLs in the rule tree outside of the system branch.

1. Before updating your rule tree, export your current rule tree with custom legacy rules into a file (for example, **C:\MyRuleTrees\Acme_rule_tree.xml**) by choosing **File→Export**.

2. In Access Manager, import the new default rule tree with the updated system branch stub into Access Manager by choosing **File→Import**.

   The new default rule tree resides in *TC_ROOT***\data\tc_am_rule_tree.xml**.

3. Open the rule tree you exported in Step 1 into an XML editor and manually add the new system branch stub.

   You now have a rule tree that includes both your customizations and the updated system branch stub.

4. Import your new updated rule tree .xml file from Step 3 into Teamcenter using the **am_install_tree** command. For example:

   ```
   am_install_tree –u=Tc-admin-user –p=password –g=group
           -path=C:\MyRuleTrees\Acme_rule_tree.xml -replace_all
   ```

5. Log on to Teamcenter and verify that your Access Manager rule tree contains your customizations and the updated system branch.

Note:

If you encounter any problems with your rule tree, you can restore it as follows:

```
am_install_tree -u=Tc-admin-user -p=password -g=group
    -path=%TC_ROOT%\data\tc_am_rule_tree.xml -mode=replace_all
    -format=xml
```

Then, repeat Step 3 and Step 4 until your results are correct.

6. Access Manager automated test harness

Advantages of automating rules testing

Prior to releasing Teamcenter into production, it is a best practice to test the rules in the rule tree. This ensures users have access to data and are able to perform tasks. It also ensures no users are granted access to data or can perform a task that they should not perform. Testing the access for any change in configuration or change to the rule tree can be time-consuming, as this is generally a manual process using such tests as:

- Reviewing users logging on with user ID, group, and role.

- Finding objects.

- Performing actions, such as modify, delete, checkout, revise, and export.

- Verifying whether or not these actions are to be granted or denied.

Use the **am_rule_test_harness** utility to perform automated rules testing.

Overview of AM rule harness testing

The **am_rule_test_harness** utility automates rules testing with minimal configurations, thereby reducing time, expense, and errors.

To use the **am_rule_test_harness** utility, you must:

1. Define search criteria in a test **input XML file**, which specifies the user, group, role combination, object, and privileges to be tested.

    > Note:
    >
    > - The format for the search criteria is:
    >
    >   *className{attrb1=value1,attrib2=value2...}*
    >
    > - Only single-value attributes, including those from parent classes, are supported.
    >
    > - The following special characters cannot be used in class name, attribute name, or attribute value: **{ } =,**.
    >
    > - Wildcard characters are supported and defined by the **TC_pattern_match_style** preference.
    >
    > - Attribute value for the date range must be in the following format:

```
creation_date=\"start-date to end-date.
```

For example, to specify objects created from 01 June to 20 June:

```
creation_date=\"01-Jun-2016 00:00 to 20-Jun-2016 04:00.
```

- For the input XML file, **user_id**, **group**, and **role** values are mandatory. Only the **project** value is optional.

2. Run the **am_rule_test_harness** utility. When the utility is run, it searches for the specified objects and evaluates whether the privileges are granted or denied for the given user, group, and role combination.

3. Review the output XML file. If the generated output report indicates corrections need to be made, correct the data in the input file, and rerun the tests using the updated input file.

4. If the generated output report indicates corrections need to be made, correct the data in the input file, and rerun the tests using the updated input file.

Sample XML files

Sample input XML file

The **am_rule_test_harness** utility requires an input XML file, which specifies the user, group, role combination, object, and privileges to be tested.

Following is a sample input XML file:

```
<TestSuite name="Test Suite 1" description="This test suite will test x,y,z">

    <UserTest description="Tests READ for Manufacturing group on Items is denied" user_id="user2"
    group="Manufacturing" role="Designer" project="" searchCriteria="Item{item_id=id1}">
        <PrivilegeTest privilege="READ" expectedResult="Deny"/>
        <PrivilegeTest privilege="WRITE" expectedResult="Deny"/>
        <PrivilegeTest privilege="DELETE" expectedResult="Grant"/>
        <PrivilegeTest privilege="CICO" expectedResult="Deny"/>
        <PrivilegeTest privilege="EXPORT" expectedResult="Deny"/>
        <PrivilegeTest privilege="IMPORT" expectedResult="Deny"/>
    </UserTest>

    <UserTest description="Tests for Engineering group on Items" user_id="user2"
    group="Engineering" role="Analyst" project="" searchCriteria="Item{item_id=id1}">
        < PrivilegeTest privilege="READ" expectedResult="Deny"/>
        < PrivilegeTest privilege="WRITE" expectedResult="Deny"/>
        < PrivilegeTest privilege="DELETE" expectedResult="Grant"/>
        < PrivilegeTest privilege="CICO" expectedResult="Deny"/>
        < PrivilegeTest privilege="EXPORT" expectedResult="Deny"/>
        < PrivilegeTest privilege="IMPORT" expectedResult="Deny"/>
    </UserTest>
</TestSuite>
```

Sample output XML file

When the **am_rule_test_harness** utility is run with an input XML file, the utility generates an output report, for example:

```xml
<TestSuite name="Test Suite 1" description="This test suite will test x,y,z" >

  <Test description="Tests READ for Manufacturing group on Items is denied" user_id="user2"
      group="Manufacturing" role="Designer" password="password" project=""
      searchCriteria="Item{item_id=001}">

    <object searchCriteria="Item{item_id=001}">

      <Privilege value="READ" expectedResult="DENY" status="Pass"
      AM_Rule_Path="Has Class(POM_application_object)/Has Class(POM object)"
      Named_ACL="Working" Accessor_type="World"/>

      <Privilege value="WRITE" expectedResult="DENY" status="Pass" AM_Rule_Path="
      Has Class(POM_application_object)/Has Class(POM object)" Named_ACL="Working"
      Accessor_type="World"/>

      <Privilege value="DELETE" expectedResult="GRANT" status="Pass" AM_Rule_Path="
      Has Class(POM_application_object)/Has Class(POM object)" Named_ACL="Working"
      Accessor_type="World"/>

      <Privilege value="CICO" expectedResult="DENY" status="Pass" AM_Rule_Path="
      Has Class(POM_application_object)/Has Class(POM object)" Named_ACL="Working"
      Accessor_type="World"/>

      <Privilege value="Export" expectedResult="GRANT" status="Pass" AM_Rule_Path="
      Has Class(POM_application_object)/Has Class(POM object)" Named_ACL="Working"
      Accessor_type="World""/>

      <Privilege value="Import" expectedResult="DENY" status="Pass" AM_Rule_Path="
      Has Class(POM_application_object)/Has Class(POM object)" Named_ACL="Working"
      Accessor_type="World"/>

    </object>

    <object searchCriteria="Item{item_id=001}">
```

If the generated output report indicates corrections need to be made, correct the data in the input file and rerun the tests using the updated input file.

Perform automatic rules testing

1. Write an **input XML file** that defines your search criteria.

    > Note:
    >
    > When the Access Manager rule tree contains the **Current Group Is** condition, the **am_rule_test_harness** utility uses the group from the current logged-on user and not the group specified in the input XML file.

2. Run the **am_rule_test_harness** utility. For example:

```
am_rule_test_harness -u=johnadmin -p=passjohn -g=dba
  -inputFile=C:\inputDir\am_rule_test_harness_input.xml
  -outputDir=C:\output
```

3. Review the **output XML file**.

4. If necessary, troubleshoot test errors using the Access Manager Test Harness report. Correct data in the input XML file and rerun the tests.

Additional ways to manage data

There are different types of administration data, for example, Access Manager rules and Organization data. At times, it is necessary to manage administrative data by moving data between your development and production environments. Because administration data is locally owned, moving this data between sites is handled differently from shared data.

To ensure proper operation, both sites should share the same Teamcenter version. However, if both sides have the same data model for the data being exchanged, the exchange can occur with different versions of Teamcenter and still operate properly.

You can use Teamcenter Environment Manager (TEM) to manage your administration data at multiple sites. For example, you can export and import administration data using panels in TEM that are accessed through the **Manage Administration Data** option in the **Feature Maintenance** panel. Using TEM, you can select the specific instances of administration data by category, class, and specific attribute/value criteria.

You can do the following:

*   Generate and view a report containing Access Manager administration data using the **generate_admin_data_report** utility.

*   Generate and view a report comparing administration data at two sites using the **generate_admin_data_compare_report** utility.

- Export Access Manager named ACLs and privileges using the **admin_data_export** utility.

- Import Access Manager administration data using the **admin_data_import** utility.

7. Verifying the effect of access rules

About verifying the effect of access rules

After you implement access rules, verify that the rules produce the desired privileges for different types of accessors. You can do this by viewing the access privileges in My Teamcenter. You can also determine which rules result in a privilege being granted or denied by viewing the verdicts in the **Extra Protection** dialog box.

In addition, you can view performance statistics.

Determining access privileges

View access privileges

Use the **Access** dialog box to determine the access privileges you have to an object. You can also view the access privileges for another user.

> Note:
>
> For quick access to summary access information, you can use the Information Center in the lower portion of the Teamcenter window next to the clipboard.

1. In My Teamcenter, select the object affected by the access rule and choose **View→Access**.

   > Tip:
   >
   > You can also right-click the object and choose **Access** from the shortcut menu, or you can click **Access** 🔒 on the toolbar.

   The **Access** dialog box appears, showing the privileges that the logged-on user has to the selected object.
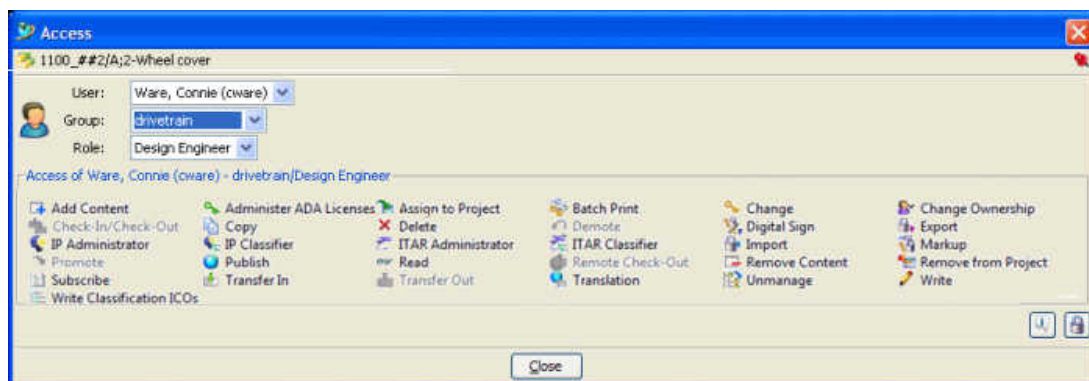
2. To view privileges assigned to your other roles and groups, select the role or group from the lists in the **Access** dialog box.

   The system updates the **Access** table to reflect the privileges of the selected group and role.

3. To view the privileges of a different user, select the user, group, and role from the lists in the **Access** dialog box.

   The system updates the **Access** table to reflect the privileges of the selected user, group, and role.

View access privileges example

In this example, you see privileges for two users for one object.

1. To view access on a selected object, choose **View→Access**.

   The **Access** dialog box shows the user **taylor** has several privileges, such as **Delete**, **Read**, and **Write** privileges to the **000017/A** item.
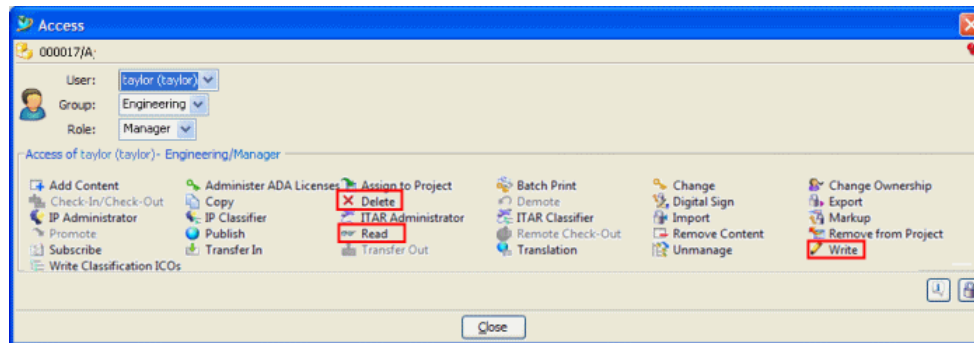
2. To view the privileges of a different user, select the user, group, and role from the lists in the **Access** dialog box.
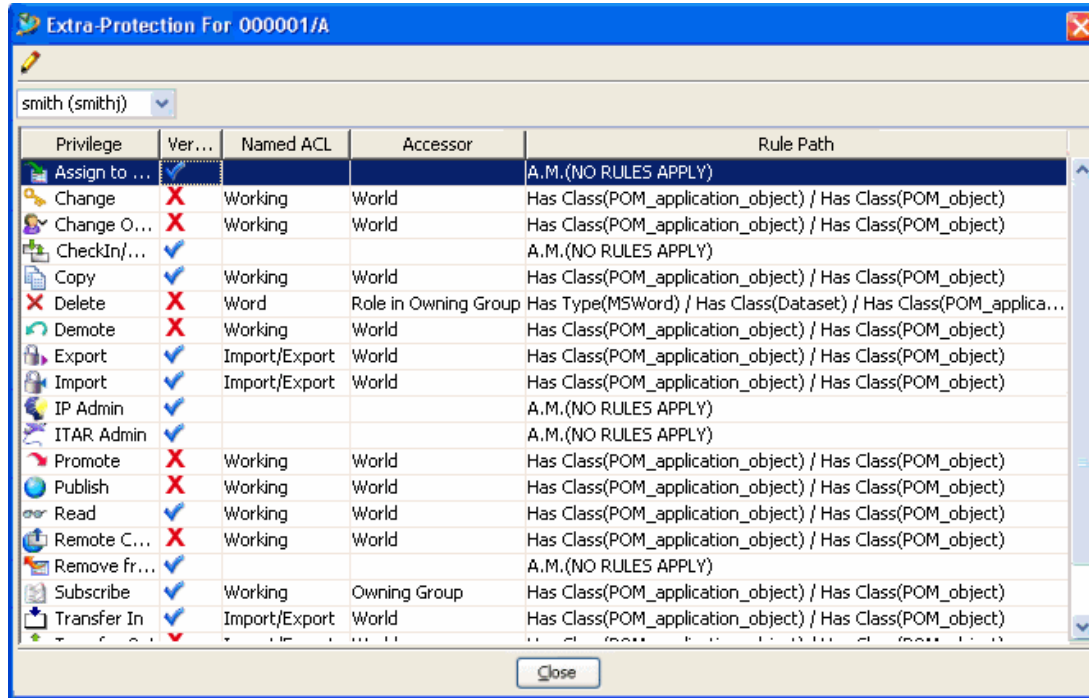
   The **Access** dialog box shows the user **smith** has **Read** and **Write** privileges but does not have **Delete** privileges to the **000017/A** item.
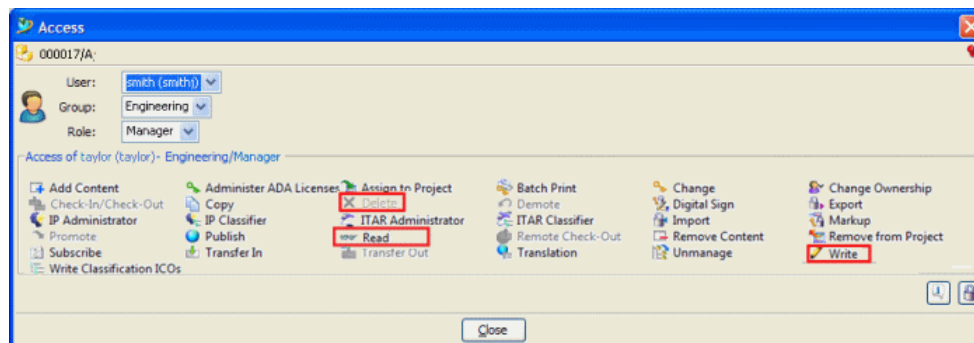
View the rules from which privileges are derived

- In the **Access** dialog box, click **Display extra protection** .
  The **Extra Protection** dialog box appears, showing the rules that apply to a privilege being granted or denied.

> **Note:**
>
> The **Access** dialog box and the **Extra Protection** dialog box may display different information.
>
> - The **Access** dialog box displays information based on the current user and that user's group and role.
>
> - The **Extra Protection** dialog box displays information based on the current user, without assessing the current user's group or role.

View the access control list (ACL) associated with the object

- In the **Access** dialog box, click .
  The system displays the **ACL Control List** dialog box.
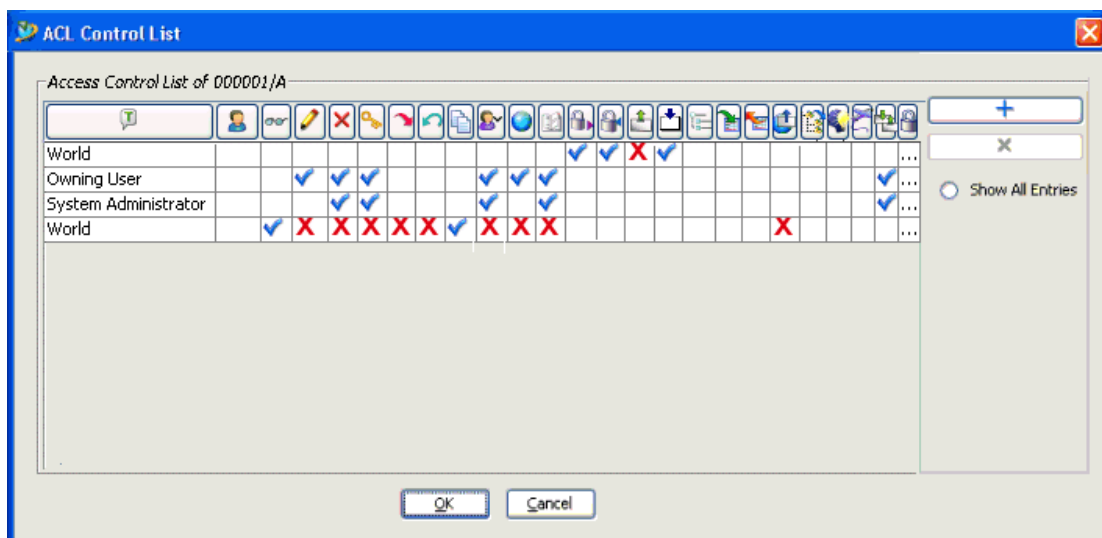
View performance statistics

You can use the **AM_PERFORMANCE_STATISTICS** environment variable to view Access Manager performance statistics for each call to a rule or accessor function. For example, if you customize access rules, you can use the performance statistics to view the performance of the customization. The statistics are logged to the **syslog** file at server shutdown.

> Note:
>
> Because there is a significant performance impact to collect the statistics, the feature is disabled by default.

Statistics are logged in both **grep**/Excel-compatible and human-readable format. The **grep** utility is used to extract the statistics entries from the **syslog** file using the **AM_STATISTIC_ENTRY** string. Each resulting entry is in comma-separated values (CSV) format for import into Microsoft Excel.

grep/Excel format:

```
AM_STATISTIC_ENTRY,entry_type,name,call_count,min_cpu,max_cpu,total_cpu,
min_real,max_real,total_real,min_sql,max_sql,total_sql
Where:
entry_type: RULE | ACCESSOR
name:       Name of the rule or accessor function
call_count: Total number of calls to this Rule or Accessor function
min_cpu:    Minimum number of seconds of CPU time used by a call to this function
max_cpu:    Maximum number of seconds of CPU time used by a call to this function
total_cpu:  Total number of seconds of CPU time used by all calls to this function
min_real:   Minimum number of seconds of real time used by a call to this function
max_real:   Maximum number of seconds of real time used by a call to this function
total_real: Total number of seconds of real time used by all calls to this function
min_sql:    Minimum number of SQL requests used by a call to this function
max_sql:    Maximum number of SQL requests used by a call to this function
```

```
total_sql:  Total number of SQL requests used by all calls to this function
```

The following is an example in **grep**/Excel using the CSV format:

```
AM_STATISTIC_ENTRY,RULE,Owning
User,8601,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0,0,0
AM_STATISTIC_ENTRY,RULE,Has
Class,198591,0.000000,0.016000,0.186000,0.000000,0.016000,0.156000,0,0,0
AM_STATISTIC_ENTRY,RULE,Has
Status,16416,0.000000,0.016000,0.031000,0.000000,0.016000,0.031000,0,0,0
AM_STATISTIC_ENTRY,RULE,In
Job,8208,0.000000,0.016000,0.016000,0.000000,0.016000,0.016000,0,0,0
AM_STATISTIC_ENTRY,ACCESSOR,World,321,0.000000,0.000000,0.000000,0.000000,0.000000,0.00
0000,0,0,0
```

The human-readable format contains the statistics in tabular form with column and row labels.

The following is an example in human-readable format:

```
Access Manager Rule Statistics

Rule_Name                                                  Total Calls
   Resource          Minimum        Maximum        Average        Total

Owning User                                                        8601
   CPU Time          0.000000       0.000000       0.000000       0.000000
   Real Time         0.000000       0.000000       0.000000       0.000000
   SQL Calls                0              0       0.000000              0
Has Class                                                        198591
   CPU Time          0.000000       0.016000       0.000001       0.186000
   Real Time         0.000000       0.016000       0.000001       0.156000
   SQL Calls                0              0       0.000000              0
Has Status                                                        16416
   CPU Time          0.000000       0.016000       0.000002       0.031000
   Real Time         0.000000       0.016000       0.000002       0.031000
   SQL Calls                0              0       0.000000              0
In Job                                                            8208
   CPU Time          0.000000       0.016000       0.000002       0.016000
   Real Time         0.000000       0.016000       0.000002       0.016000
   SQL Calls                0              0       0.000000              0

Access Manager Accessor Statistics

Accessor_Name                                              Total Calls
   Resource          Minimum        Maximum        Average        Total

World                                                              321
   CPU Time          0.000000       0.000000       0.000000       0.000000
   Real Time         0.000000       0.000000       0.000000       0.000000
   SQL Calls                0              0       0.000000              0
```