webidblack_2linecentr

**USER’S**

**MANUAL**

*Access Control Policy Tool (ACPT)*

**National Institute of Standards and Technology**

July 2018

**Revision Sheet**

|  |  |  |  |
| --- | --- | --- | --- |
| **Release No.** | **Date** | **Revision Description** | **Author** |
| Rev. 0 | 06/01/11 | Draft 1 of manual | Nathan Jacobson  SURF Student 2011 |
| Rev. 1 | 08/14/12 | Draft 2 of manual | JeeHyun Hwang  North Carolina State Univeristy |
| Rev. 2 | 06/24/13 | Draft 3 of manual | Bruce Batson  SURF Student 2013 |
| Rev.3 | 05/14/15 | Draft 4 of manual - Added five policy combination algorithms: Weak-Consensus, Strong-Consensus, Weak-Majority, Strong-Majority, and Super-Majority-Permit. | Ang Li  University of Arkansas Student |
| Rev.4 | 5/15/18 | Draft 5 of manual – Added Separation of Duty function and updated test module. | Wei Bao  University of Arkansas Student |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**USER'S MANUAL**

**TABLE OF CONTENTS**

Page #

[1.0 GENERAL INFORMATION 1-1](#_Toc519109191)

[1.1 Software Overview 1-1](#_Toc519109192)

[2.0 RULES AND CONCEPTS 2-1](#_Toc519109193)

[2.1 Subject, Resource, Action Attributes 2-1](#_Toc519109194)

[2.1.1 What is an Attribute 2-1](#_Toc519109195)

[2.1.2 What is Rank? 2-2](#_Toc519109196)

[2.1.3 What is Inheritance? 2-2](#_Toc519109197)

[2.2 Algorithms 2-3](#_Toc519109198)

[2.2.1 First Applicable 2-3](#_Toc519109199)

[2.2.2 Deny Override 2-3](#_Toc519109200)

[2.2.3 Permit Override 2-3](#_Toc519109201)

[2.2.4 Weak-Consensus 2-4](#_Toc519109202)

[2.2.5 Strong-Consensus 2-4](#_Toc519109203)

[2.2.6 Weak-Majority 2-4](#_Toc519109204)

[2.2.7 Strong-Majority 2-4](#_Toc519109205)

[2.2.8 Super-Majority-Permit 2-4](#_Toc519109206)

[2.3 ABAC 2-4](#_Toc519109207)

[2.4 MultiLevel Security Model 2-5](#_Toc519109208)

[2.4.1 No Read Up (Bell-LaPadula model) 2-5](#_Toc519109209)

[2.4.1 No Write Down (Biba model) 2-5](#_Toc519109210)

[2.5 WorkFlow 2-6](#_Toc519109211)

[2.6 Security Requirements 2-6](#_Toc519109212)

[2.7 Combination 2-7](#_Toc519109213)

[2.8 Verification 2-7](#_Toc519109214)

[2.9 Test 2-7](#_Toc519109215)

[2.10 XACML 2-7](#_Toc519109216)

[2.11 Default Deny Rule 2-7](#_Toc519109217)

[2.12 Select Policies to be Merged 2-7](#_Toc519109218)

[2.13 Combined Policies 2-7](#_Toc519109219)

[3.0 INTERFACE 3-1](#_Toc519109220)

[3.1 New project 3-1](#_Toc519109221)

[3.2 Saving Project 3-2](#_Toc519109222)

[3.3 Open Project 3-2](#_Toc519109223)

[3.4 Subject, Resources Actions - Attributes and Attribute Values Add/Update/Delete 3-3](#_Toc519109224)

[3.5 Inheritance Beneficiary and Inherited Value Add/Delete 3-7](#_Toc519109225)

[3.6 ABAC Model 3-10](#_Toc519109226)

[3.7 MultiLevel Security Model 3-14](#_Toc519109227)

[3.8 WorkFlow Model 3-19](#_Toc519109228)

[3.9 Security Requirements 3-23](#_Toc519109229)

[3.9.1 Security Requirements ABAC Example 3-23](#_Toc519109230)

[3.9.2 Security Requirements MultiLevel Example 3-26](#_Toc519109231)

[3.9.3 Security Requirements WorkFlow Example 3-29](#_Toc519109232)

[3.9.4 Separation of Duty Example 3-31](#_Toc519109233)

[3.10 Combination 3-34](#_Toc519109234)

[3.11 Verification 3-36](#_Toc519109235)

[3.11.1 Select policies to be merged 3-36](#_Toc519109236)

[3.11.2 Combined Policies 3-36](#_Toc519109237)

[3.11.3 Verification Results Examples 3-37](#_Toc519109238)

[3.12 Test 3-36](#_Toc519109239)

[3.12.1 Select policies to be merged 3-36](#_Toc519109240)

[3.12.2 Combined Policies 3-36](#_Toc519109241)

[3.12.3 Test Results Example 3-37](#_Toc519109242)

[3.13 XACML 3-38](#_Toc519109243)

[3.13.1 Select policies to be merged 3-38](#_Toc519109244)

[3.13.2 Combined Policies 3-38](#_Toc519109245)

[3.13.3 XACML File Example 3-39](#_Toc519109246)

[4.0 PROJECT REFERENCES 4-1](#_Toc519109247)

[4.1 References 4-1](#_Toc519109248)

# GENERAL INFORMATION

**1.0 General Information**

This software was developed by the National Institute of Standards and Technology (NIST) in corporation of Computer Science Department of North Carolina State University. Pursuant to title 15 United States Code Section 105, works of NIST employees are not subject to copyright protection in the United States and are considered to be in the public domain. As a result, a formal license is not needed to use the software.

This software is provided by NIST as a service and is expressly provided "AS IS". NIST MAKES NO WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT AND DATA ACCURACY. NIST does not warrant or make any representations regarding the use of the software or the results thereof including, but not limited to, the correctness, accuracy, reliability or usefulness of the software.

Permission to use this software is contingent upon your acceptance of the terms of this agreement.

## 1.1 Software Overview

The Access Control Policy Tool (ACPT) is a toolkit written in JAVA. This program is cross platform compatible. One can run this software on any machine that has Java Virtual Machine (JVM) installed. ACPT allows one to simulate different policies in a particular environment. A simulation environment is built up of three inputs: Subjects, Resources and Actions. After one establishes their environment, he or she can apply one of three models: Attribute Based Access Control Model (ABAC), MultiLevel Security Model (MLS), and WorkFlow Policy Model. The user can set up different properties and use the NuSMV verification tool to see how a specific rule complies. This software will also run your policy(s) through a series of tests powered by Advanced Combinatorial Testing System [<http://csrc.nist.gov/groups/SNS/acts/index.html>]. Once the policy is complete it can be exported to an eXtensible Access Control Markup Language (XACML) file.

# RULES AND CONCEPTS

**2.0 Rules and Concepts**

This section introduces some access control concepts related to ACPT.

## 2.1 Subject, Resource, Action Attributes

### 2.1.1 What is an Attribute

The ACPT uses attributes to help set up an environment. The environment contains the subject/users, resources and actions. Then the ACPT user will define a model and tested on that specific environment. All attributes have a name, type and attribute value(s). For example *Roles* is the subject name of type *string* with *attribute values*: *Student*, *Professor*, and *Registrar*. One can have as many attributes as desired. The ACPT only supports up to one level of classification for its attributes. An attribute is the root and it can only have one level of children for its attribute values.

**Correct**

**Incorrect**

**Attribute**:

Roles

**Attribute Value**:

Student

Attribute:

Roles

Attribute Value:

Student

Attribute describer:

Teacher\_Assistant

### 2.1.2 What is Rank?

Rank is numerical value attribute that can only be mapped to any subject or resource attributes for the MultiLevel Model. The higher the value of the integer placed on the rank attribute is more classified (in terms of Multilevel Security Level model). When creating a Rank Attribute, the Attribute name, for example, could be any text, for example “Rank\_1” and its attribute type is an Integer with the attribute value “2” for example. Note that when applying rank attributes it can only enter 1 value per rank attribute.

Attribute

Rank\_1

Attribute

Ranks

2

0

1

2

**Correct**

**Incorrect**

### 2.1.3 What is Inheritance?

Inheritance is where a subject attribute value follows the same rules as another subject attribute value. One example would be if you had two subject values, Student and Teacher. If the Teacher inherits Student and Student is permitted to read file 1, then Teacher is permitted to read file 1. There can be multiple levels of Inheritance but there cannot be a loop of Inheritance. One subject value can inherit more than one subject value.

Attribute Value

Dean

**Correct**

**Incorrect**

Attribute Value

Teacher

Attribute Value

Teacher

Attribute Value

Student

Attribute Value

Teacher

Attribute Value

Student

Attribute Value

Teacher

Attribute Value

Student

## 2.2 Algorithms

The ACPT uses three different algorithms to help the ACPT user specify the way to handle duplicate rules in the same model or if multiple models are merged together.

### 2.2.1 First Applicable

First applicable takes the permission decision of the first occurrence of a particular variable if the same variable was in the policy twice with a different Permission. For example if two policies were merged, it would take the permission decision of the first one.

### 2.2.2 Deny Override

Deny Override scans through the entire model to find all the duplicate variables. Once a deny permission of duplicate variable is found it will deny the access to that particular rule in the model.

### 2.2.3 Permit Override

Permit Override scans through the entire model to find all the duplicate variables. Once a grant permission of duplicate variable is found it will permit the access to that particular rule in the model.

### 2.2.4 Weak-Consensus

Weak-consensus requires that policies should not conflict with each other. It denies an access request if some policies deny the request, and no policy permits it. It permits an access request if some policies permit the request, and no policy denies it. It outputs conflict if some policies permit and some deny.

### 2.2.5 Strong-Consensus

This algorithm requires that all policies must agree on a decision. It denies an access request if all policies deny the request. It permits an access request if all policies permit the request. Conflict is output otherwise. Note that this algorithm is different from weak-consensus since a policy may neither permit nor deny a request (e.g., it might not be applicable to the request). When some policies deny a request and others are not applicable to it, weak-consensus denies the request but strong consensus outputs conflict.

### 2.2.6 Weak-Majority

When different policies make conflicting decisions (permit and deny) about a request, the request is permitted (denied, resp.) if the number of policies permitting (denying, resp.) it is greater than the number of policies denying (permitting, resp.) it.

### 2.2.7 Strong-Majority

Strong-majority permits (denies, resp.) a request if more than half of all policies permit (deny, resp.) it.

### 2.2.8 Super-Majority-Permit

Super-majority-permit permits an access request if more than 2/3 of all policies permit it, and denies it otherwise.

## 2.3 ABAC

The ABAC model template allows ACPT users to define access control rules based on their environment variables (Subject, Resource, and Action). Unlike MLS and Workflow, ABAC template consists rules without any embedded access control model.

Example:

Subject: Student, Professor, Registrar

Resource: Grades, Transcript  
Action: Read, Read and add note, and Write

Rules:

1. Subject: Student; Resource: Grade; Action: read; Permit
2. Subject: Student; Resource: Grade; Action: write; Deny
3. Subject: Student; Resource: Grade; Action: modify; Deny
4. Subject: Student; Resource: Transcript; Action: read; Permit
5. Subject: Student; Resource: Transcript; Action: write; Deny
6. Subject: Student; Resource: Transcript; Action: modify; Deny
7. Subject: Professor; Resource: Grade; Action read; Permit
8. Subject: Professor; Resource: Grade; Action write; Permit
9. Subject: Professor; Resource: Grade; Action: modify; Permit
10. Subject: Professor; Resource: Transcript; Action: read; Permit
11. Subject: Professor; Resource: Transcript; Action: write; Deny
12. Subject: Professor; Resource: Transcript; Action: modify; Deny
13. Subject: Registrar; Resource: Grade; Action: read; Permit
14. Subject: Registrar; Resource: Grade; Action: write; Deny
15. Subject: Registrar; Resource: Grade; Action: modify; Deny
16. Subject: Registrar; Resource: Transcript; Action: read; Permit
17. Subject: Registrar; Resource: Transcript; Action: write; Permit
18. Subject: Registrar; Resource: Transcript; Action: modify; Permit

## 2.4 MultiLevel Security Model

The MultiLevel Security Model (MLS) enforce Bell-LaPadula and Biba models, which protect resources from being accessed from unauthorized ranking members. This particular model uses rule properties: No read up, No write down. These properties ensure no subject cannot read a resource above their access level or write to a resource lower than their current rank.

### 2.4.1 No Read Up (Bell-LaPadula model)

This property handles, which subjects can have access to read the resource. Any subject can read their current ranked resource and any resource below their current rank. It is forbidden to read any resource above its rank.

**For Example:** No Read Up

Subject: Resource:

2 2

1 1

0 0

Note: There are three users ranked 0, 1, and 2, and three resources ranked 0, 1, and 2. Higher the rank of the subject or resource higher the security level it has.

### 2.4.1 No Write Down (Biba model)

This property protects information from being changed from unauthorized subjects. A subject can write to a resource as long as they are the same rank level or above their current rank. However, a higher ranked subject cannot write to a lower ranked resource. Higher ranked subjects have access to more classified information so if they are not allowed to write down this will help prevent information leaking to unauthorized subjects.

**For Example:** No Write Down

Subject: Resource:

2 2

1 1

0 0

Note: There are three users ranked 0, 1, and 2, and three resources ranked 0, 1, and 2. Higher the rank of the subject or resource higher the security level it has.

## 2.5 WorkFlow

The WorkFlow model enforces access privileges by sequential states. A state is the access from the subject to a resource at any given order. Each state can only permit or deny one rule before moving to the next.

Example: Shop

Subject: Roles → Client, Engineer, Builder

Resource: Classifier → Blueprint  
Action: MLSDefaultAction→ Read, Read and add note, Write

State 1: Client → Blueprint → Read and add note

State 2: Engineer → Blueprint → Read

State 3: Engineer → Blueprint → Writes

State 4: Builder → Blueprint → read

Note: Access is denied for all subjects at all states unless specified.

## 2.6 Security Requirements

The Security Requirements tab allows the ACPT users to design a specific access case to verify against rules and model entered through ABAC, MLS, and Workflow tabs. This tab can also allow configuration for Separation of Duty. Section 3.9 demonstrates an example.

## 2.7 Combination

The Combination tab allows the ACPT user to select models created from model tabs and order by priorities. This step prepares the models before they can be combined for verification and testing. Section 3.10 demonstrates this operation.

## 2.8 Verification

The Verification tab allows the ACPT user to test his/her security requirements specified in the Security Requirements tab against the model.

## 2.9 Test

The Test tab will run your policy(s) through a series of tests powered by Advanced Combinatorial Testing System [<http://csrc.nist.gov/groups/SNS/acts/index.html>].

## 2.10 XACML

The eXtensible Access Control Markup Language (XACML) tab will generate a XACML file for a model designed in the ACPT. XACML is structure of Extensible Markup Language (XML). The XACML specializes in writing rules to a file for Access Control Systems.

## 2.11 Default Deny Rule

This is an option throughout the ACPT software. When this feature is selected the model will deny access to any rule that is not specified in the policy. This feature can be found on the verification, test and XACML tabs.

## 2.12 Select Policies to be Merged

When “Selecting Policies to be Merged,” it appends each policies after each other.

## 2.13 Combined Policies

When combining policies it ranks the policies in order of highest priority from selected list in the “Combined” tab. The user has the option to choose one of the algorithms (Section 2.2) to handle the duplicate entries.

# Interface

**3.0 Interface**

In this section, the ACPT user can see visual diagrams of step-by-step overview of how to use the ACPT Software.

## 3.1 New project

When the software opens the interface is grayed out until the user creates a new project.

Go to **File** → **New Project**   
Short cut: **Ctrl-N**

|  |  |
| --- | --- |
| newProject  Figure 3.1.1 ACPT start up selecting New Project | Name your project in the dialog box    Figure 3.1.2 ACPT naming you project |
| Figure 3.1.3 ACPT main default window after creating new project |  |

## 3.2 Saving Project

If this is your first time saving your project, go to **File** → **Save As** or **File** → **Save**

You may use the shortcut Ctrl-S after you have saved your project initially

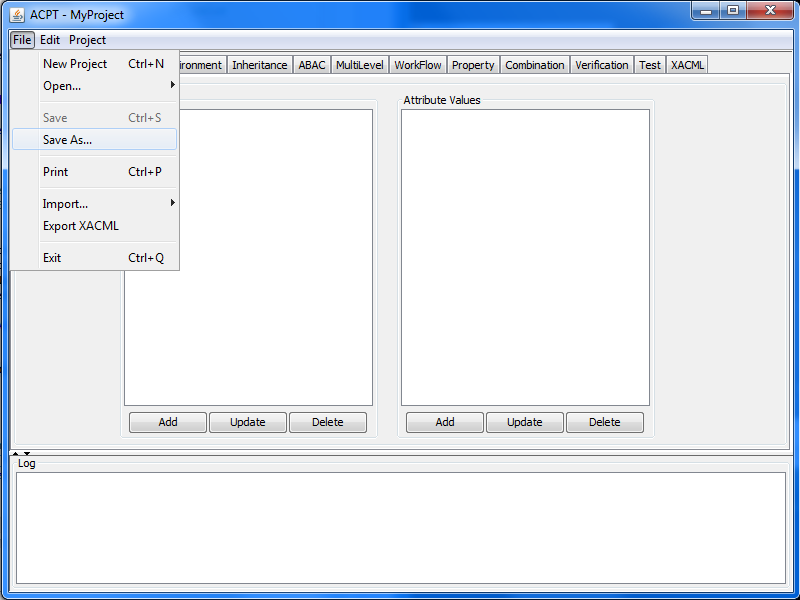
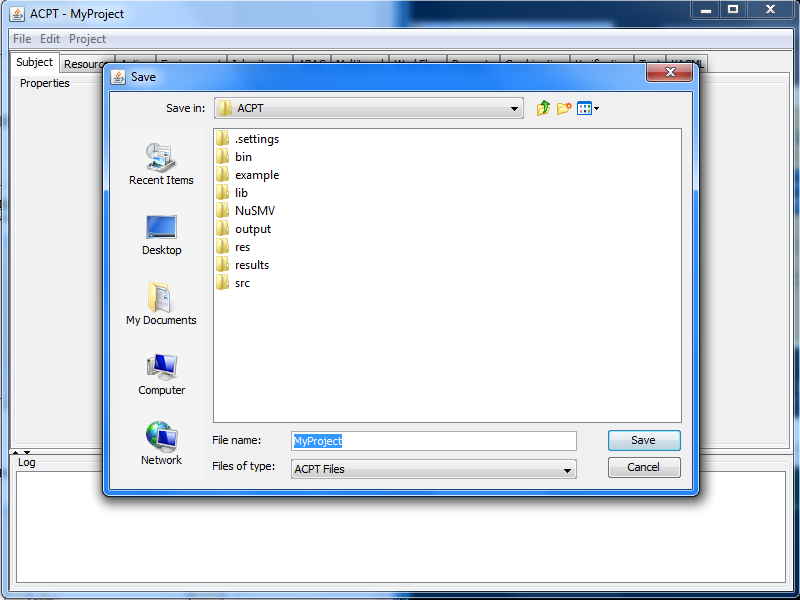


Figure 3.2.1 ACPT selecting Save As Figure 3.2.2 ACPT Save dialog box Example Program

## 3.3 Open Project

To open previous projects, go to **File** → **Open** → **Project**  
Shortcut: **Ctrl → O**

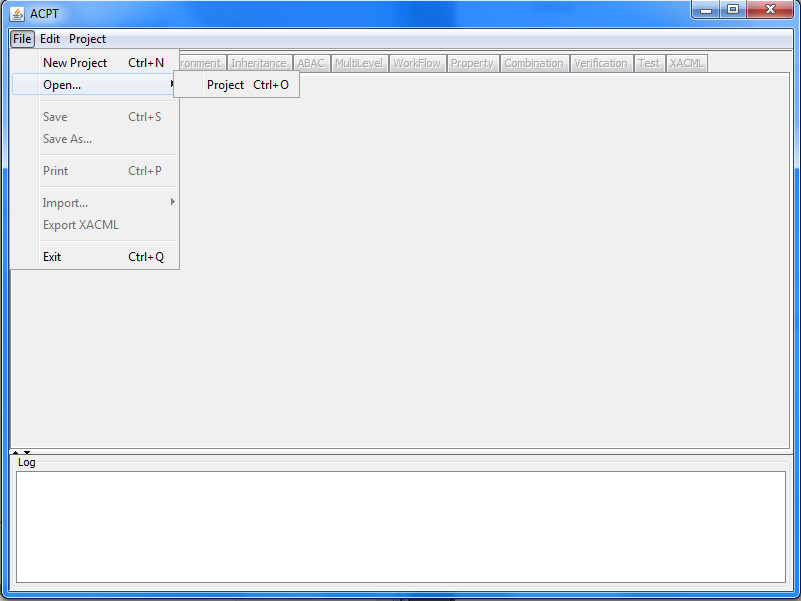
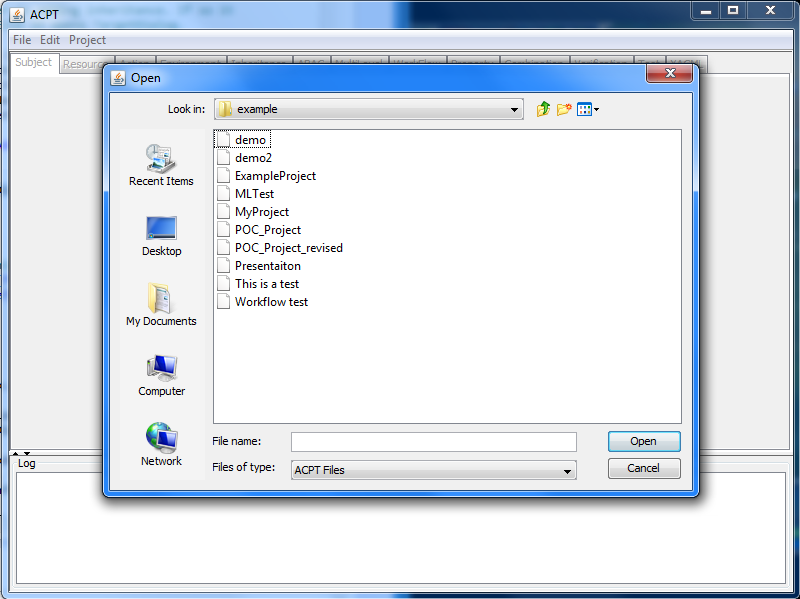


Figure 3.3.1 ACPT select File → Open → Project Figure 3.3.2 Select a file from dialog box

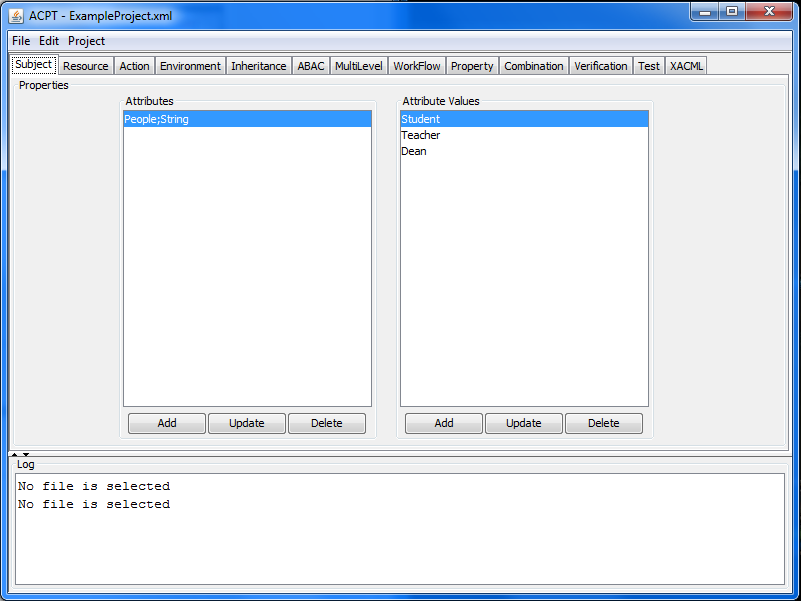
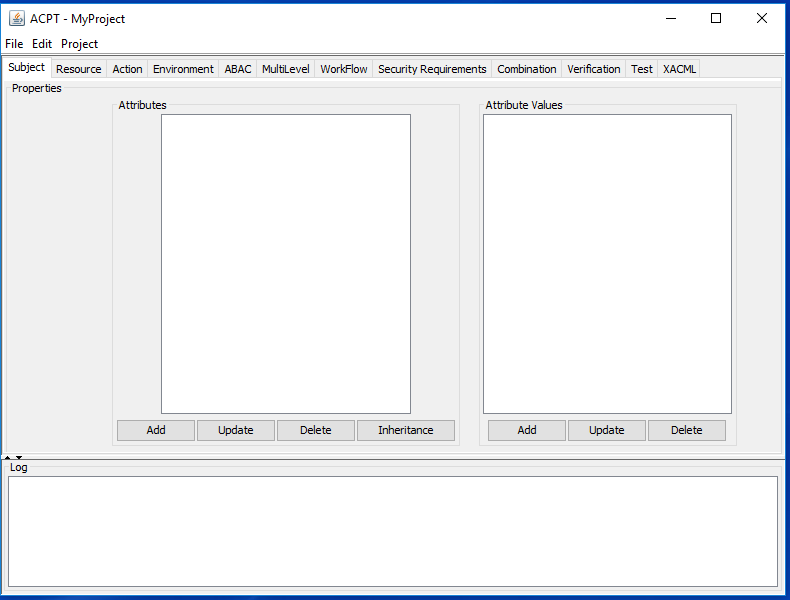


Figure 3.3.3 ACPT main window of selected file

## 3.4 Subject, Resources Actions - Attributes and Attribute Values Add/Update/Delete



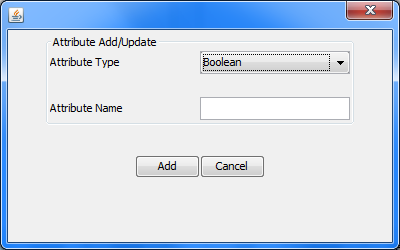
Attributes

Attribute Values

Figure 3.4.1 ACPT Subject tab

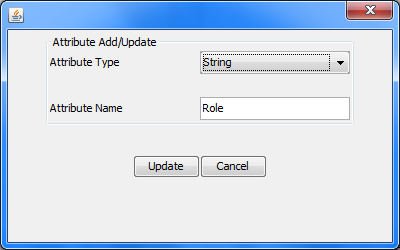
**Attributes**

To add an Attribute:

1. Click **Add** from the “Properties” field on the left side of the screen. A box will pop up  
     
   Figure 3.4.2 New attribute window
2. Select your attribute type from the drop down menu: Boolean, Integer or String.
3. Name the attribute
4. Click **Add**

To update an Attribute:

1. Click **Update** from the “Properties” field on the left side of the screen. A box will pop up

  
Figure 3.4.3 Update attribute window

1. Change the attributes name.
2. Click **Update**

Note: You can only change the name of the selected attribute if the selected attribute has at least one

Attribute Value. If an attribute is equipped to a policy you may not **Update** it

To Delete an Attribute:

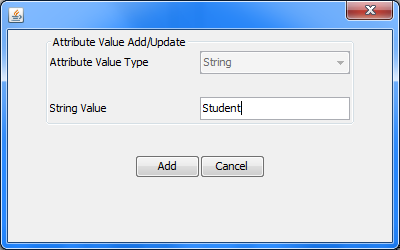
1. Select the attribute that needs to be deleted on the list from the “Properties” field on the left side of the screen.
2. Click **Delete**

Note: If an attribute is equipped to a policy. The attribute cannot be deleted because that policy will not be complete and will fault. This is one security check for the ACPT.

**Attribute Values**

An Attribute value can only be added after and equipped to an attribute.

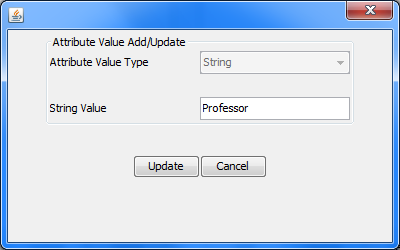
To Add an Attribute Value

1. Select the appropriate Attribute on the left side of the Properties field.
2. Click **Add** on the right side of the **Properties** field. A Box will pop up  
     
     
   Figure 3.4.4 New attribute value window
3. Give it a unique String name. Example: *Student*
4. Click **Add**

Note: Attribute type is grayed out because you selected the type when you created the Attribute.  
The following caveats are applied when using the ACPT:  
   
1. An attribute name (subject, action, resource) should be started with only letters, and the name can only contains letters, and underscore. This limitation is due to the model checking engine used for the ACPT.  
2. To specify properties (in the Property tab), every attributes in a specified properties need to be in the range of predefined attributes values, for example, if a subject attribute Students has values: Sue, John, and Dave. And a specified property contains: "SPEC (Students = Bob)..." The verification result will generate error report complaining that the value Bob is "undefined".

To update an Attribute Value:

1. Click **Update** from the **Properties** field on the right side of the screen. A box will pop up

  
Figure 3.4.5 Update attribute value window

1. Change the String Value. Example: *Student* is now *Professor*
2. Click **Update**

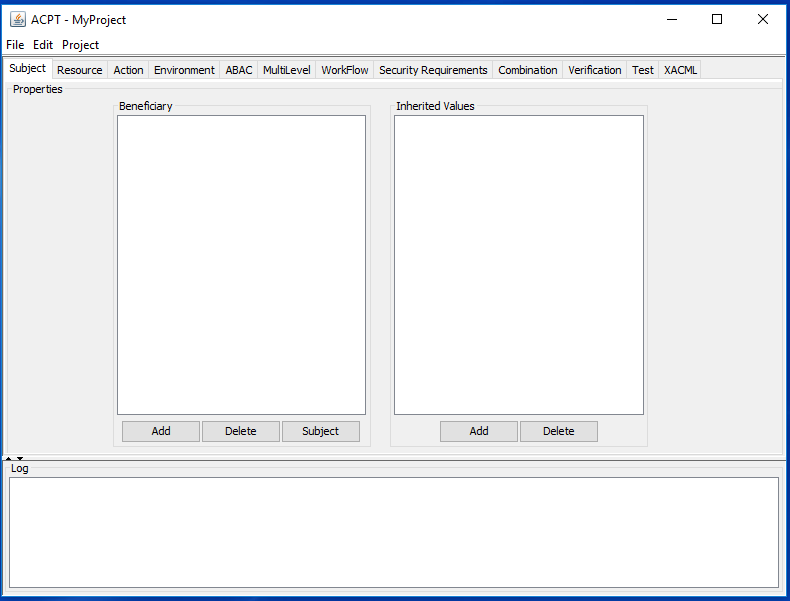
Note: If an Attribute Value is equipped to a policy you may not **Update** it.

To Delete an Attribute:

1. Select the attribute value that needs to be deleted on the list from the **Properties** field on the right side of the screen.
2. Click **Delete**

Note: If an attribute value is equipped to a policy. The attribute cannot be deleted because that policy will not be complete and will fault. This is one security check for the ACPT.

## 3.5 Inheritance Beneficiary and Inherited Value Add/Delete



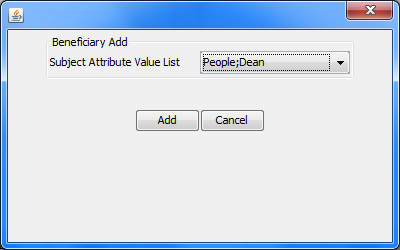
Inherited values

Beneficiary

Figure 3.5.1 ACPT Inheritance panel

**Beneficiary**

To add a Beneficiary:

1. Click **Add** from the “Properties” field on the left side of the screen. A box will pop up  
    Figure 3.5.2 New beneficiary window
2. Select your Beneficiary from the drop down menu
3. Click **Add**

Note: If a Beneficiary is add that is used in a policy, the policy will not be updated to add inheritance

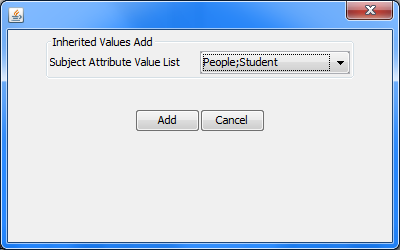
To Delete a Beneficiary:

1. Select the Beneficiary that needs to be deleted on the list from the “Properties” field on the left side of the screen.
2. Click **Delete**

Note: If a Beneficiary value is equipped to a policy. The attribute cannot be deleted because that policy will not be complete and will fault. This is one security check for the ACPT.

**Inherited value**

To add a Inherited value:

1. Click **Add** from the “Properties” field on the left side of the screen. A box will pop up  
    Figure 3.5.2 New inherited value window
2. Select your Inherited value from the drop down menu
3. Click **Add**

Note: If a Inherited value is add that is used in a policy, the policy will not be updated to add inheritance

To Delete an Inherited value:

1. Select the Inherited value that needs to be deleted on the list from the “Properties” field on the left side of the screen.
2. Click **Delete**

Note: If an Inherited value is equipped to a policy. The attribute cannot be deleted because that policy will not be complete and will fault. This is one security check for the ACPT.

## 3.6 ABAC Model

To apply the ABAC Model to your environment go to the **ABAC** tab on the navigation bar.

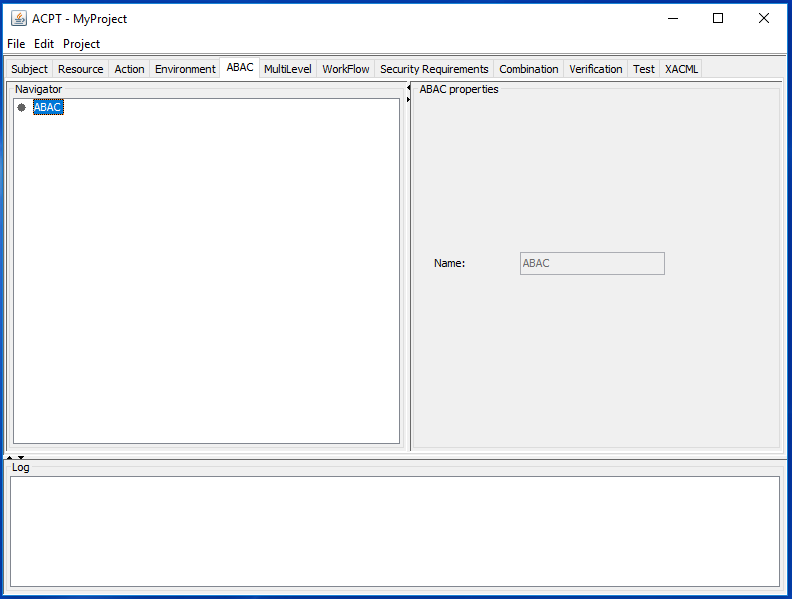
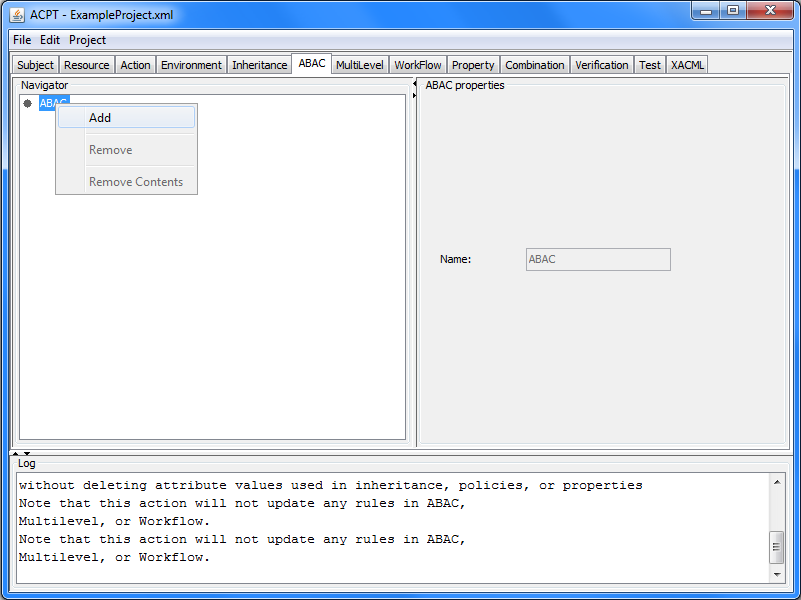


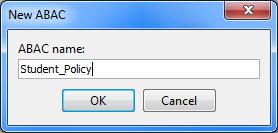
Figure 3.6.1 ACPT ABAC model tab

Create a policy using **ABAC**:

1. Highlight **ABAC** in the Navigator window
2. Right click and hit **Add**

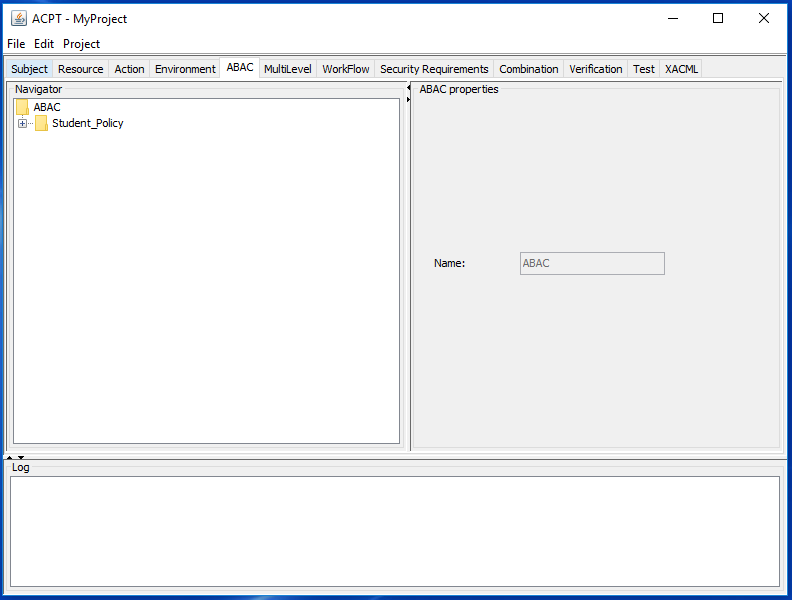
  
Figure 3.6.2 Adding a new ABAC model

1. A box will appear and give your policy a name For Example *Student\_Policy.*

  
Figure 3.6.3 Naming the ABAC model

1. Click **OK.**

Once a policy has been created the **ABAC** tab should look similar to the following screen shot.

  
Figure 3.6.4 ABAC tab with new model selected

There are three different algorithms that can be applied to **ABAC**

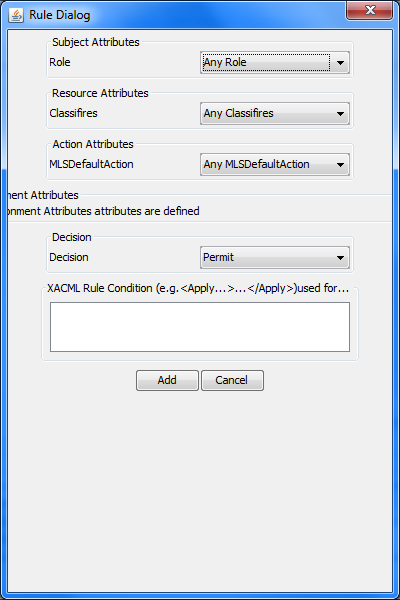
* **First Applicable**
* **Deny-overrides**
* **Permit-overrides**

To change the algorithm type:

1. Select the algorithm from the dropdown menu from **Rules combination Algorithm**
2. Click **Update**

Add a rule to your police:

1. Click **Add** from the **Rules** field. A box will popup

  
Figure 3.6.5 ABAC rule dialog box

1. For the first three fields **Subject Attributes**, **Resource Attributes**, **and Action Attributes**, It lists all the attributes one has entered from the corresponding tab. The left side says the attribute and then there is a dropdown box next to with its Attribute Values.
2. The Decision field has two options: **Permit**, **Deny**.
3. Select the decision whether you want the subject to have access or not
4. Click **Add**.

## 3.7 MultiLevel Security Model

To apply the MultiLevel Security Model to your environment go to the **MultiLevel** tab on the navigation bar.

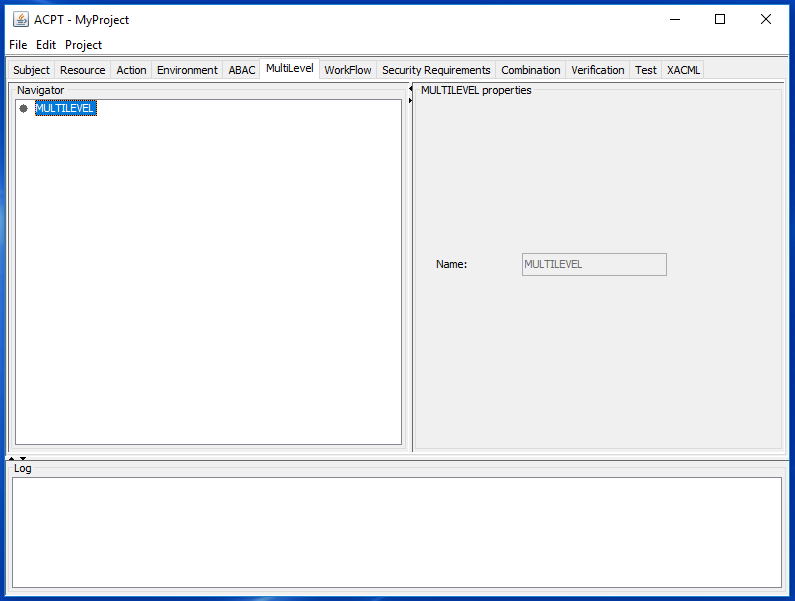


Figure 3.7.1 MultiLevel model tab

Create a policy using **MultiLevel**

1. Highlight **MULTILEVEL** in the Navigator window
2. Right click and hit **Add**

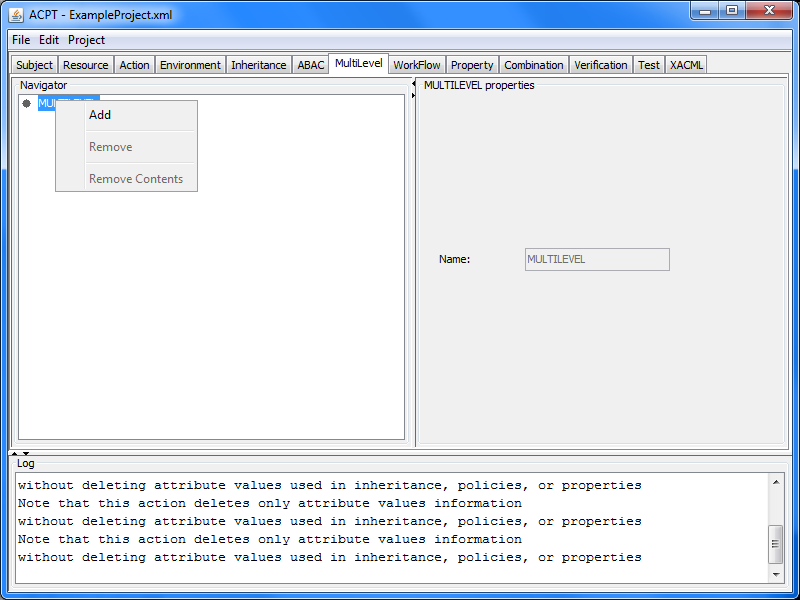


Figure 3.7.2 Adding a new MultiLevel model

1. A box will appear and give your policy a name For Example “*ARMY\_Policy*”

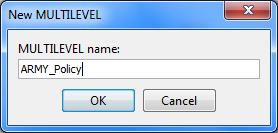
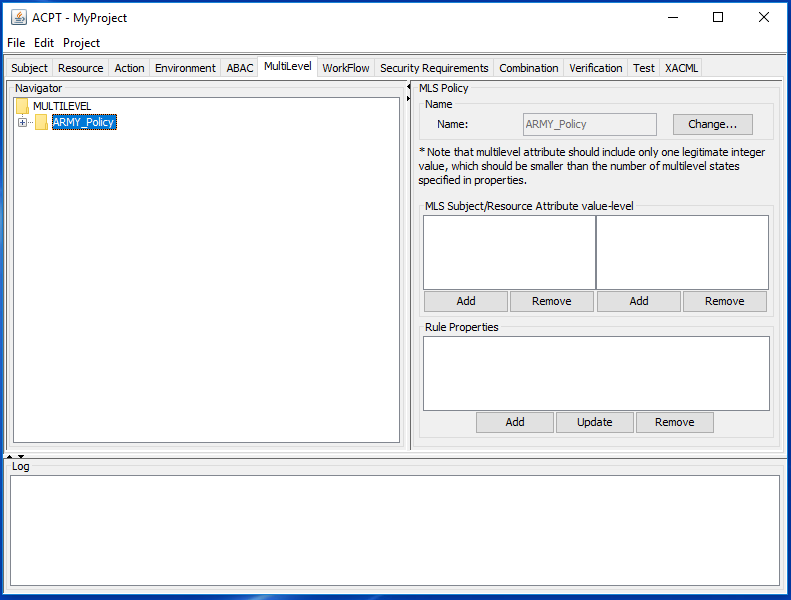


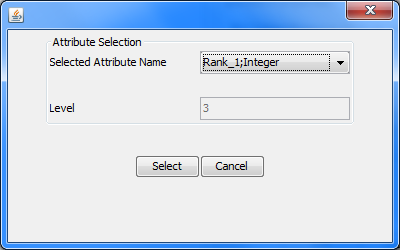
Figure 3.7.3 Naming the MultiLevel model

1. Click **OK**

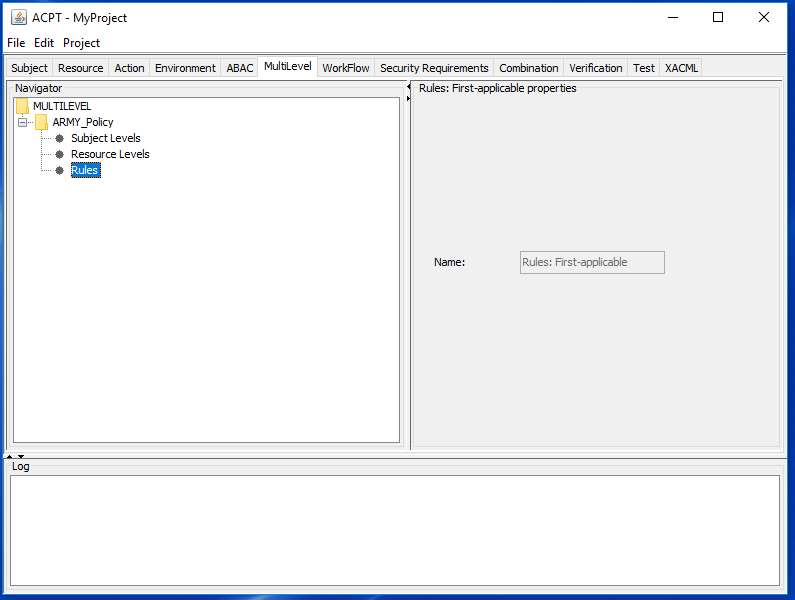
Once a policy has been created the **MultiLevel** tab should look similar to the following screen shot.  
  
  
Figure 3.7.4 MultiLevel tab with new model selected

Now **Add** ranks to both **Subject** and **Resource** Attribute value-level (same process just for different attributes) example for Subject

1. Click **Add**
2. A box will popup

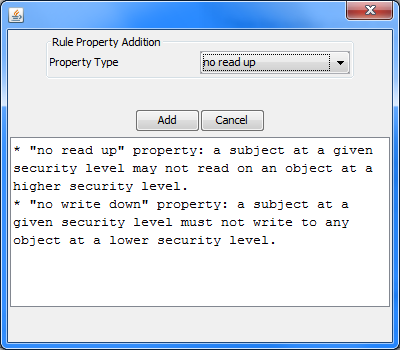
  
Figure 3.7.5 Add ranking attributes

1. Select an attribute from the drop down menu
2. Click **Select**

  
Figure 3.7.6 multiLevel tab window

How to apply either “no read up” or “no write down” Rule properties to the MultiLevel model

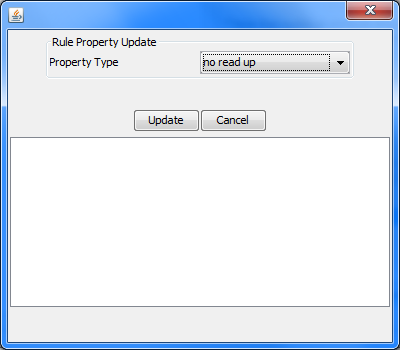
1. In the Rule Properties field, click **Add**
2. A popup box will appear

  
Figure 3.7.7 Selecting a rule property window

1. Select either **no read up** or **no write down** from the drop down menu
2. Click **add**

How to update a current Rule Property

1. Select the rule property you want to change from the **Rule Properties** field.
2. A popup box will appear

  
Figure 3.7.8 Changing rule property window

1. Select your **Property Type**
2. Click **Update**

How to Remove / Delete a Rule Property

1. Select the Rule Property that needs to be removed from the **Rule Property** field
2. Click **Remove**

## 3.8 WorkFlow Model

To apply the WorkFlow Model to your environment, go to the **WorkFlow** tab on the navigation bar.

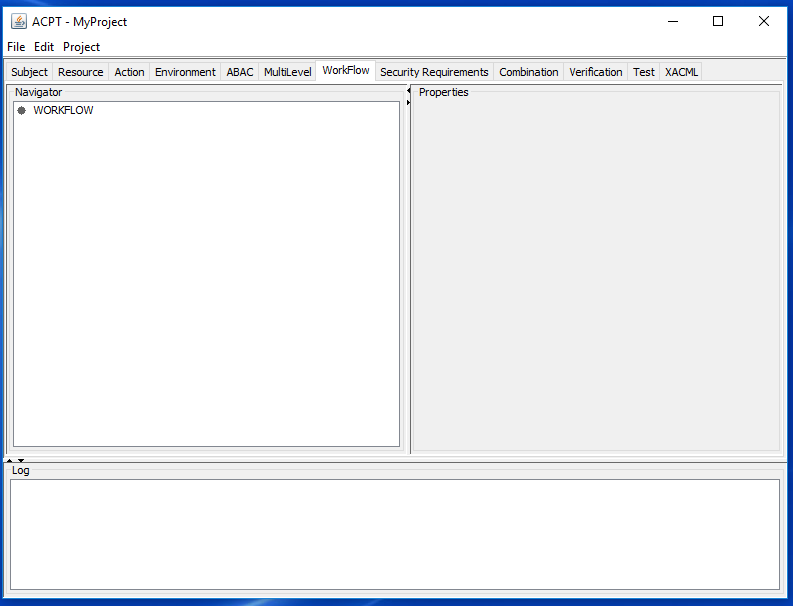


Figure 3.8.1 WorkFlow model tab

Create a policy using WorkFlow

1. Highlight **WORKFLOW** in the Navigator window
2. Right click and hit **Add**

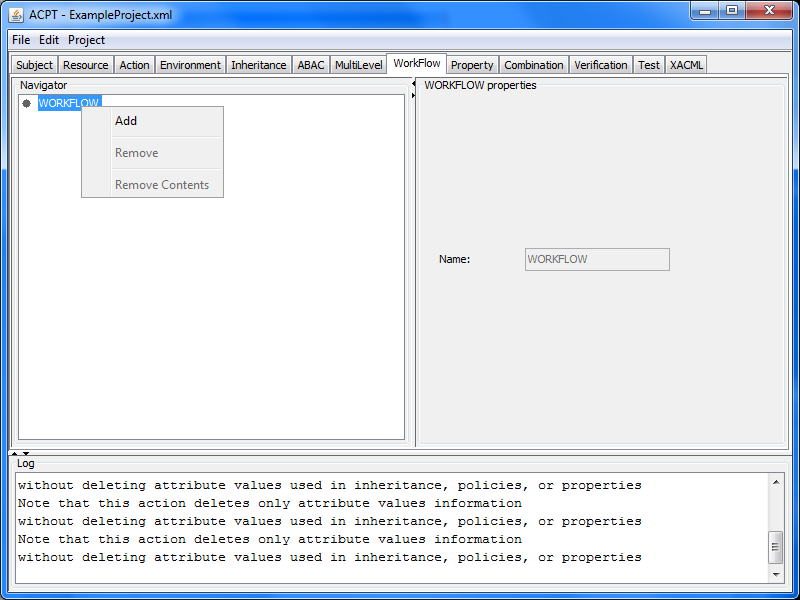
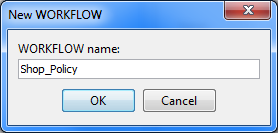
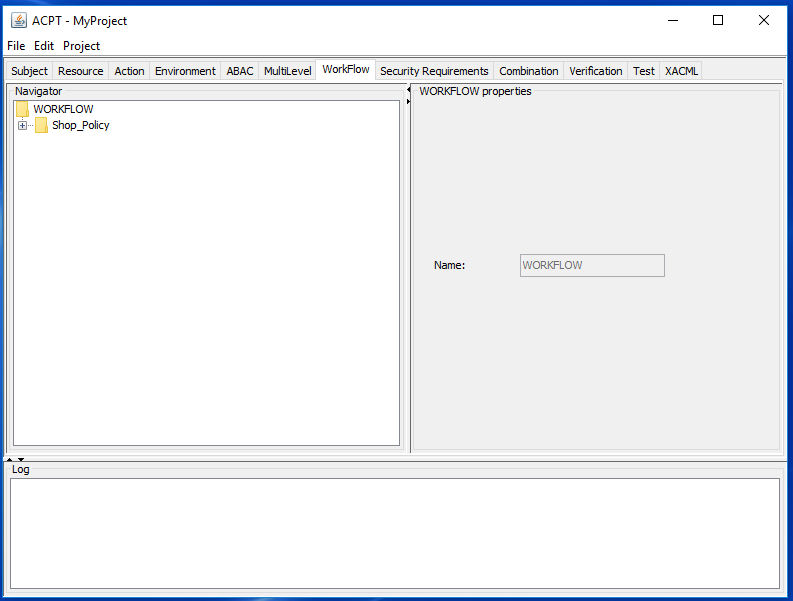


Figure 3.8.2 Adding a new WorkFlow model

1. A box will appear and give your policy a name For Example “*Shop\_Policy*”

  
Figure 3.8.3 Naming the WorkFlow model

1. Click **OK**

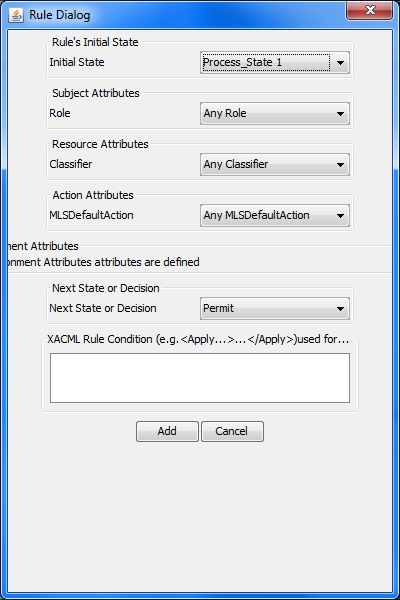
Once a policy has been created the WorkFlow tab should look similar to the following screen shot.  
  
Figure 3.8.4 WorkFlow tab with new model selected

Select a Rule Combination Algorithm

1. Go to the **Rule Combination Algorithm** field
2. Select algorithm from the dropdown menu
3. Click **Update**

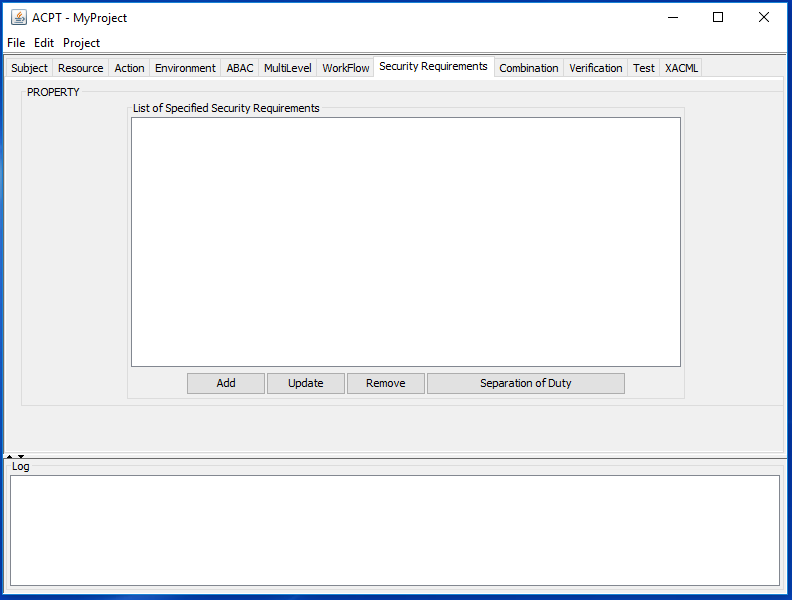
Add a Rule

1. Click **Add** button from the Rules field
2. A popup box will appear

  
Figure 3.8.5 WorkFlow rule dialog box

1. Select the **State** that the rule will apply to within the policy
2. Select Attributes for Subject, Resource, and Action
3. Select the **Decision or Next State**
4. Click **Add**

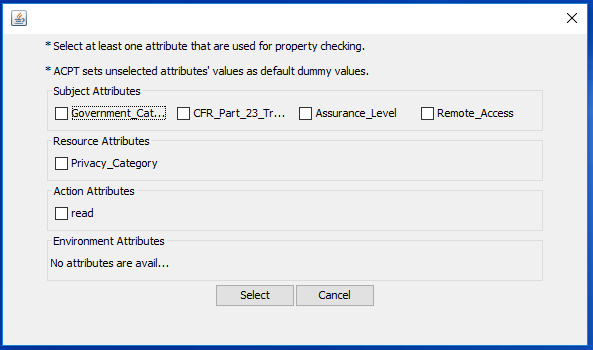
## 3.9 Security Requirements

  
Figure 3.9.1 Security Requirements tab

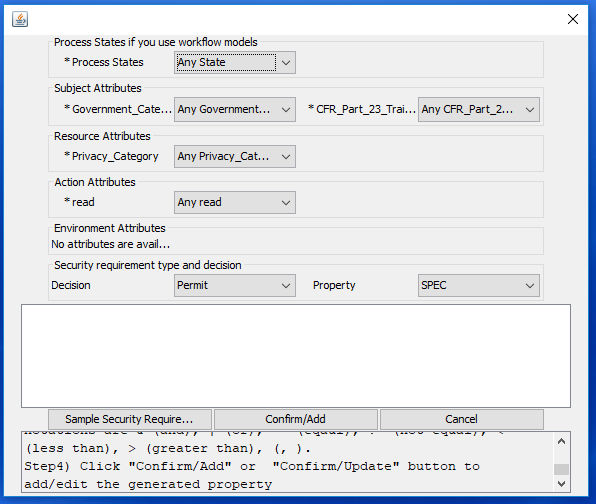
### 3.9.1 Security Requirements ABAC Example

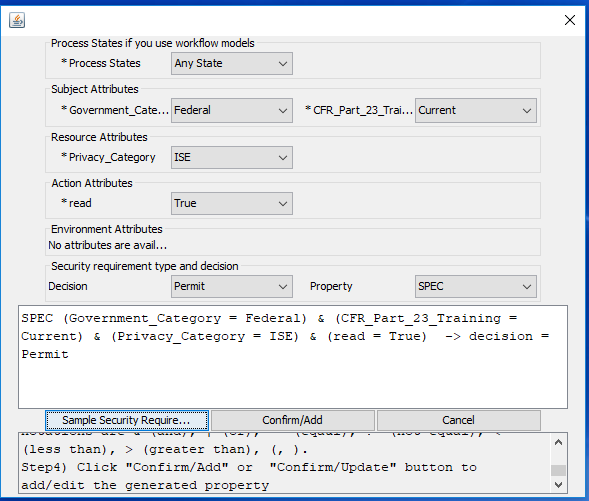
How to add a security requirement

1. Click **Add** from the List of Specified Security Requirements field
2. A popup box will appear

  
Figure 3.9.1.1 Select security requirements to be used

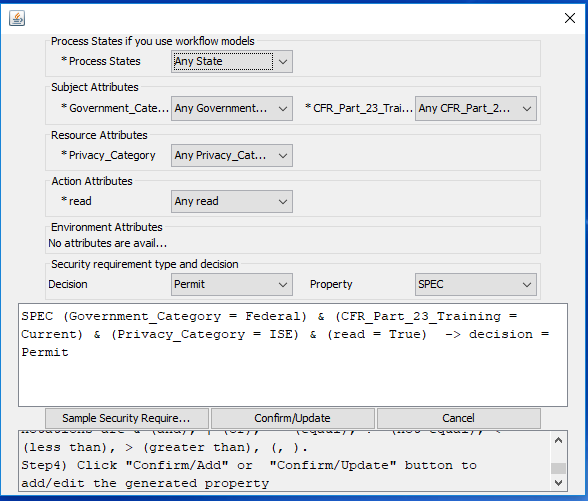
1. Select all the attributes that will be used in the security requirement checking
2. Click **Select**
3. A new popup box will appear

  
Figure 3.9.1.2 Select security requirement to be tested window

1. Select an attribute from the Subject Attribute, Resource Attribute, Action Attribute and the **Security requirement type** and **decision** fields
2. Click **Sample Security Requirement** and edit from the sample for desired security requirements.  
     
     
   Figure 3.9.1.4 Sample Security Requirement screen shot
3. Click **Confirm/Add**

How to Update a security requirement

1. Click **Update** from the List of Specified Security Requirements field
2. A popup box will appear

  
Figure 3.9.1.5 Update security requirement window

1. Select an attribute from the Subject Attribute, Resource Attribute, Action Attribute and the security requirement type and decision fields
2. Click **Sample Security Requirement** and edit it to the desired security requirements
3. Click **Confirm/Update**

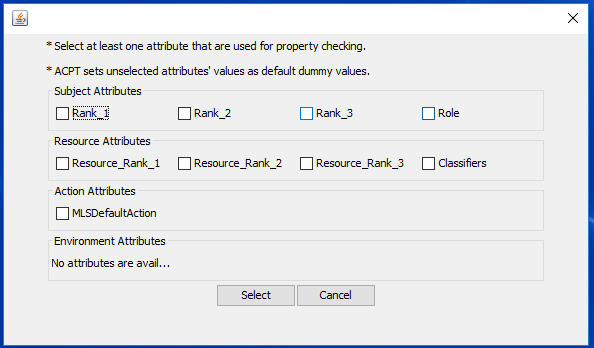
How to Remove/Delete a security requirement

1. Select the security requirement to be removed from the List of Specified Security Requirements
2. Click **Remove**.

### 3.9.2 Security Requirements MultiLevel Example

How to add a security requirement

1. Click **Add** from the List of Specified Security Requirements field
2. A popup box will appear

  
Figure 3.9.2.1 Select Security Requirement to be used

1. Select one **Rank** and **Roles** under the Subject Attributes field
2. Select one **Rank** and **Classifiers** under the Resource field
3. Select **MLSDefaultAction** under the **Action** Attributes
4. Click **Select**
5. A new popup box will appear

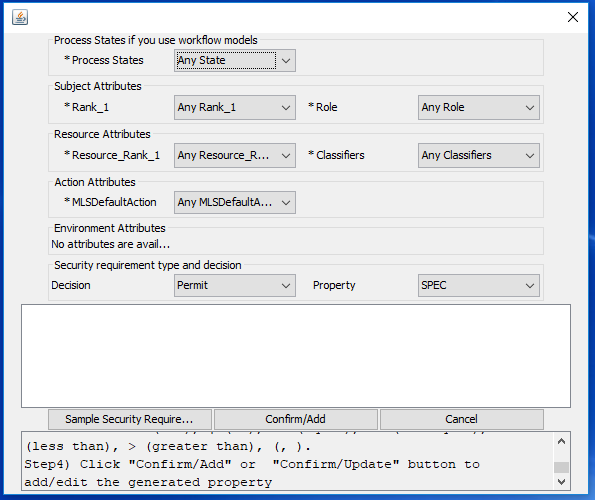


Figure 3.9.2.2 Select security requirement to be tested

1. Select attributes from the Subject Attribute, Resource Attribute, Action Attribute and the security requirement type and decision fields

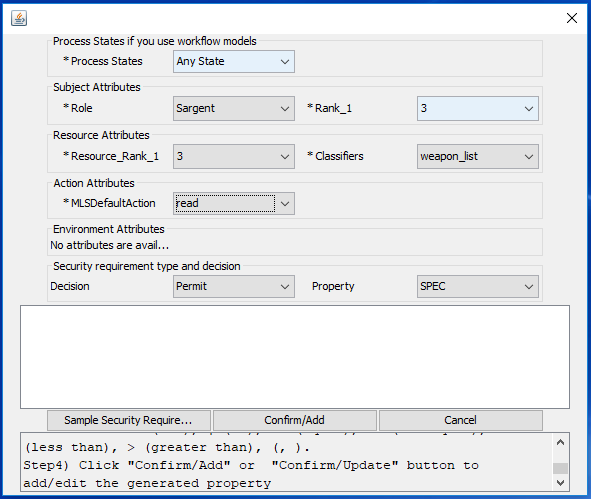


Figure 3.9.2.3 test case for model

1. Click **Sample Security Requirement** and edit to the desired security requirements.

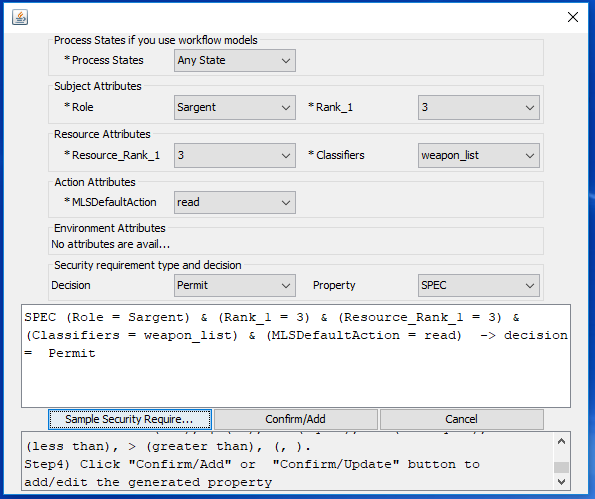


Figure 3.9.2.4 Sample Security Requirement screen shot

1. Click **Confirm/Add**

How to Update a security requirement

1. Click **Update** from the List of Specified Security Requirements field
2. A popup box will appear

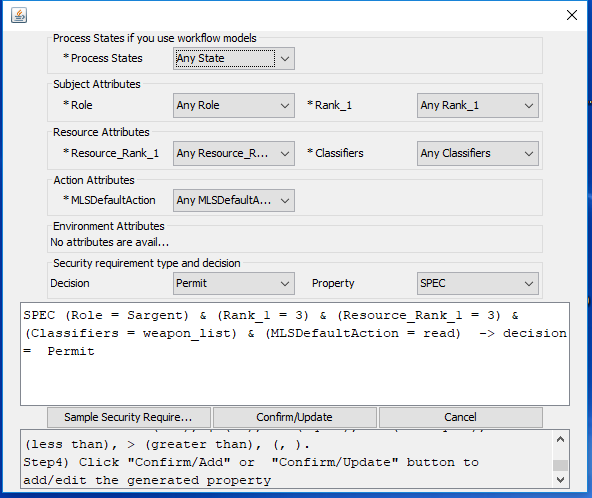


Figure 3.9.2.5 Update security requirement window

1. Select attributes from the Subject Attribute, Resource Attribute, Action Attribute and the security requirement type and decision fields
2. Click Sample Security Requirement and edit to the desired security requirements
3. Click **Confirm/Update**

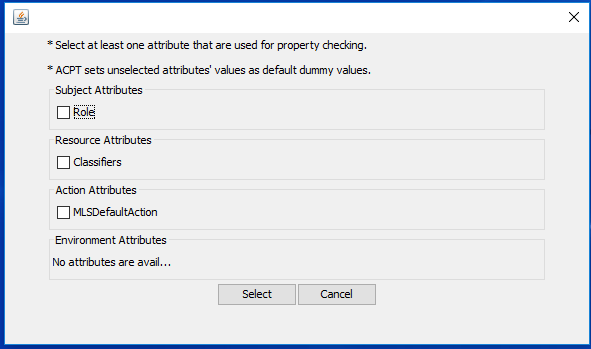
How to Remove/Delete a security requirement

1. Select the security requirement to be removed from the List of Specified Security Requirements field
2. Click **Remove**

### 3.9.3 Security Requirements WorkFlow Example

How to add a security requirement

1. Click **Add** from the List of Specified Security Requirement field
2. A popup box will appear

  
Figure 3.9.3.1 Select security requirement to be used

1. Select all the attributes that will be used in the property checking
2. Click **Select**
3. A new popup box will appear

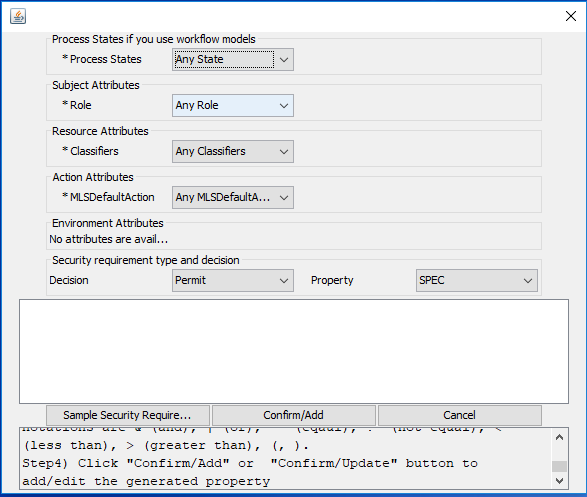


Figure 3.9.3.2 Select security requirement to be tested

1. Select **a Process State** from the **Progress States** if use workflow model field
2. Select attributes from the Subject Attribute, Resource Attribute, Action Attribute
3. Select the security requirement type and decision fields
4. Click **Sample Security requirement** and edit to the desired security requirements
5. Click **Confirm/Add**

### 3.9.4 Separation of Duty Example

The main functionality of Separation of Duty is to define several exclusive sets of combination of subjects, resources, actions, and environment.

How to select multiple sets of separation of duty combinations:

* 1. Click **Separation of Duty** to activate the separation of duty panel

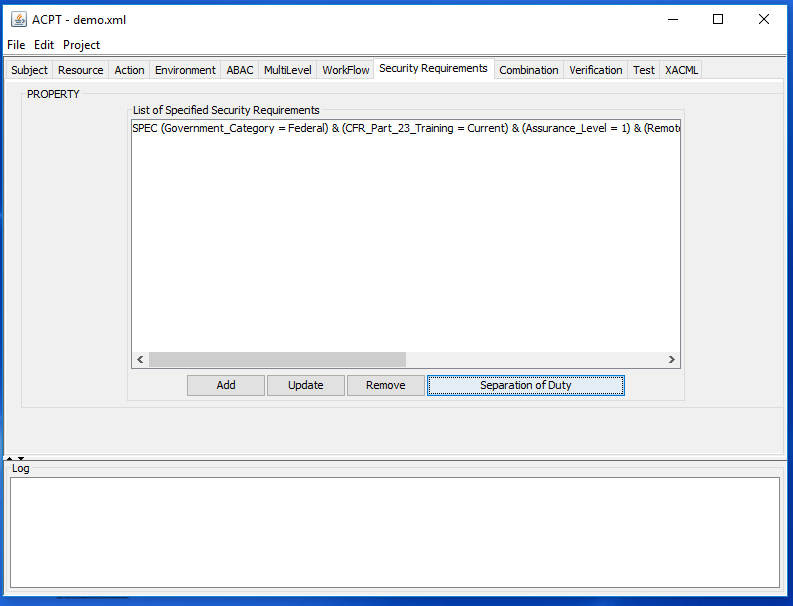


Figure 3.9.4.1 Security Requirements panel

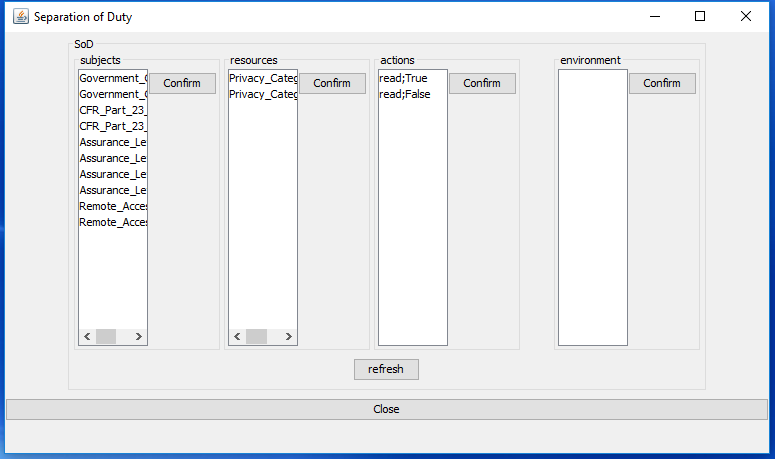


Figure 3.9.4.2 Separation of Duty panel

* 1. Select the desired subject from **subjects, resources, actions,** and **environment** list

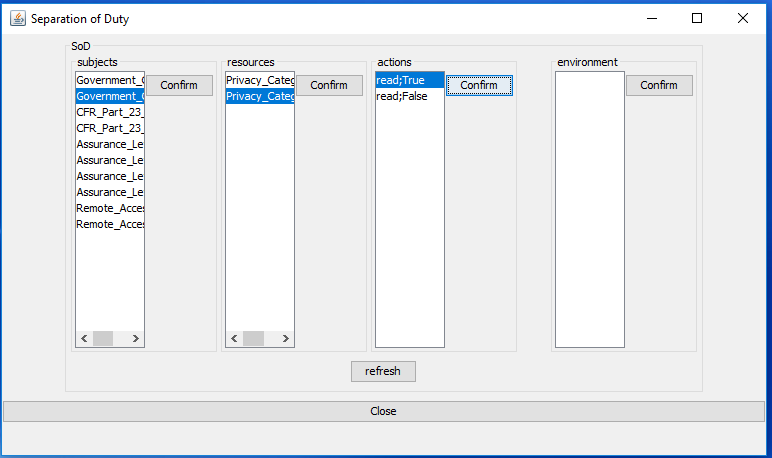


Figure 3.9.4.2 Select subjects in SoD

* 1. Click **Confirm** to confirm the selection for each category
  2. Click **refresh** to undo all the selection
  3. Click **Close** to close the window and complete one set of separation of duty combinations
  4. Redo step **1-5** to add as many sets of separation of duty combinations as needed

## 3.10 Combination

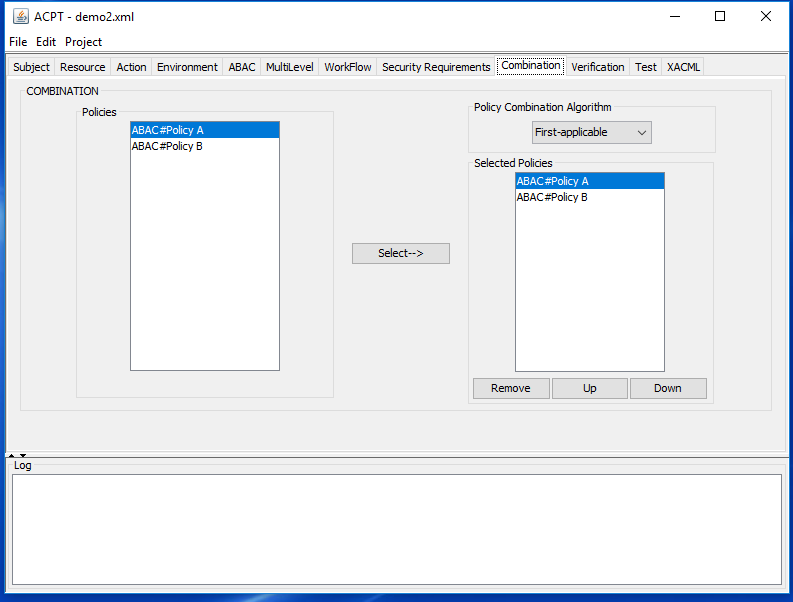


Figure 3.10.1 Combination tab

To simulate a test run using multiple policies on a first applicable combination:

1. Select policy from the **Policies** field
2. Click **Select**

Note: the policy will appear in the **Selected Policy** field

Rank your policies in the correct order

1. **Up** button: moves the select policy up one position in the list
2. **Down** button: moves the select policy down one position in the list

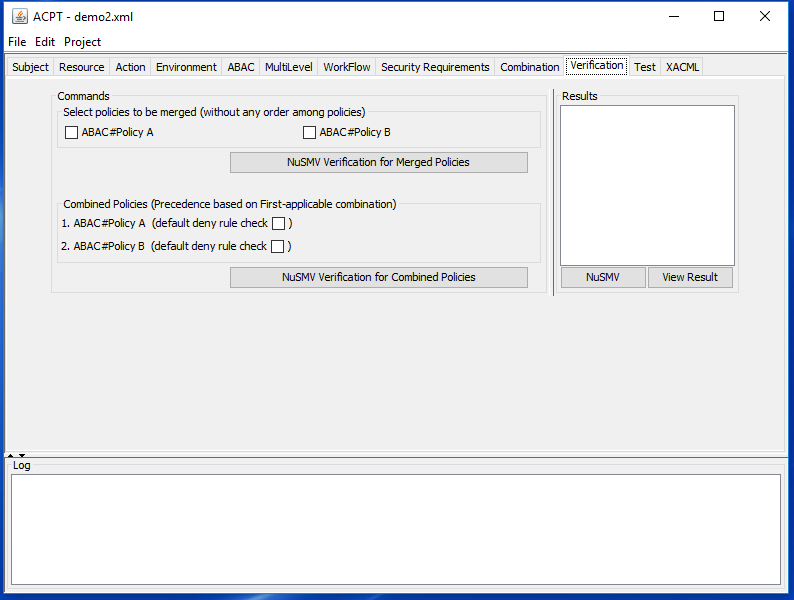
Remove a policy from the Selected Policies

1. Select a policy to be removed from the field Selected Policies
2. Click **Remove**

Choose algorithm from the Policy Combination Algorithm field

1. Select algorithm from dropdown menu
2. Click **Update**

## 3.11 Verification

 Figure 3.11.1 Verification tab

### 3.11.1 Select policies to be merged

Run a set of Security Requirements with Selected policies to be merged

1. Select the check boxes for your Security requirements to test against the specific policy
2. Click **NuSMV Verification** for Merging Policies
3. Click **View Results** in the Results field for the verification result

### 3.11.2 Combined Policies

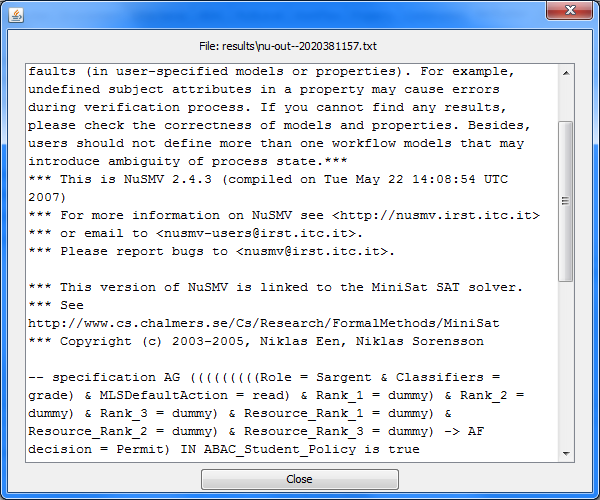
Run a set of Security Requirements with Combined Policies

1. Click **NuSMV Verification for Combined Policies**
2. Click **View Results** in the Results field for the verification result

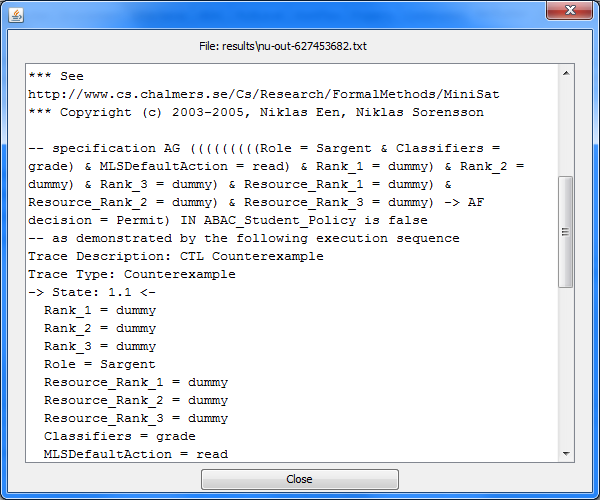
Note: The selected list from the previous tab **Combination** will be placed in the **Combined Policy** field in the same order. Optional: if checked, the default deny rule will deny access to anything not specified in the policy.

### 3.11.3 Verification Results Examples

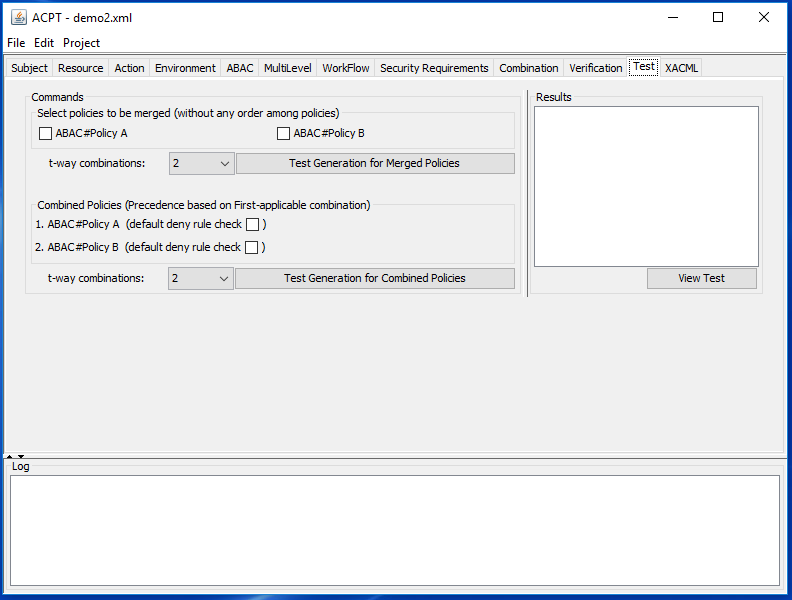
Testing the security requirement: ((Roles = Professor & Classifiers = Grades)& MLSDefaultAction = write) → decision = true, against the ABAC\_Edu\_Complete\_Policy

  
 Figure 3.11.3.1 ACPT Verification Result True

Testing the security requirement: ((Roles = Student & Classifiers = Transcript)& MLSDefaultAction = write) → decision = true, against the ABAC\_Edu\_Complete\_Policy

  
 Figure 3.11.3.2 ACPT Verification Result False (Counterexample)

## 3.12 Test

  
Figure 3.12.1 Test tab

### 3.12.1 Select policies to be merged

Generate test cases with Selected policies to be merged

1. Select the check boxes for the policies to be tested
2. Choose **t-way combination** from the dropdown menu for desired t-way
3. Click **Test Generation** for Merging Policies
4. Click **View Test** for the Results

### 3.12.2 Combined Policies

Generate test cases with Combined Policies

1. Click **Test Generation for Combined Policies**
2. Click **View Test** for the Results

Note: The selected list from the previous tab **Combination** will be placed in the Combined Policy field in the same order. Optional: if checked, the default deny rule will deny access to anything not specified in the policy

### 3.12.3 Test Results Example

The results after running the ABAC#Edu\_Complete\_Policy through the Advanced Combinatorial Testing feature of the ACPT. Note that all the attribute values from **Subject**, **Resource**, and **Action** categories can be chosen as inputs for combinatorial testing.

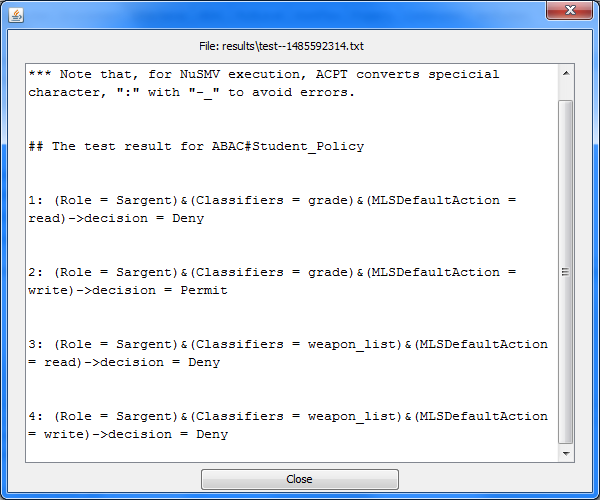


Figure 3.12.3.1 ACPT Test view

## 3.13 XACML

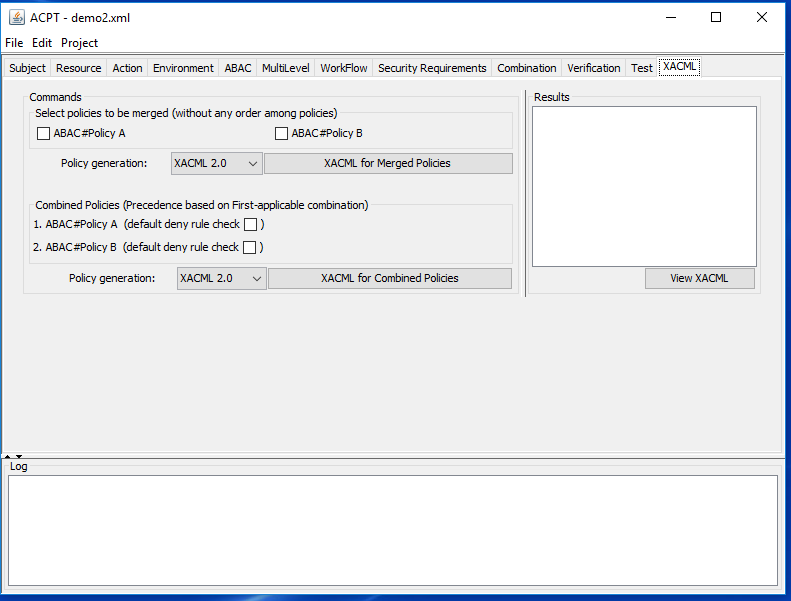


Figure 3.12.1 XACML tab

### 3.13.1 Select policies to be merged

Run selected policies to be merged to XACML

1. Select the check boxes for the policies to be merged

2. Click **XACML for Merging Policies**

3. Click **View XACML** in the Results field

### 3.13.2 Combined Policies

Run Combined Policies to XACML

1. Click **Test Generation for Combined Policies**

2. Click **View Test** in the Results field

Note: The selected list from the previous tab “Combination” will be placed in the Combined Policy field in the same order. Optional: if checked, the default deny rule will deny access to anything not specified in the policy

### 3.13.3 XACML File Example

Once a policy is created, it can be exported to a XACML file.

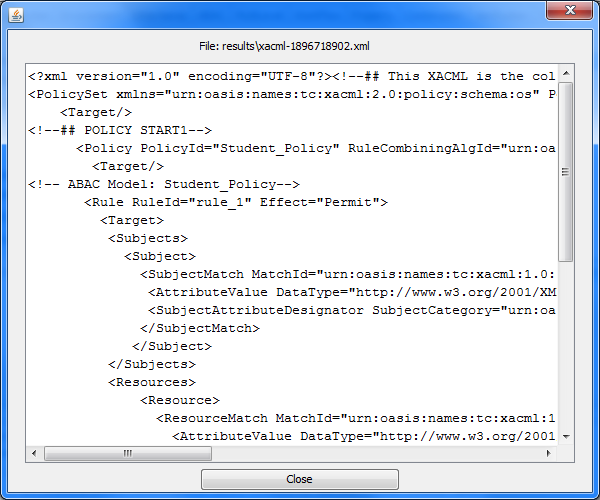


Figure 3.12.3.1 ACPT XACML File

# Project References

## 4.1 References

V.Hu “[Access Control Policy Tool](http://csrc.nist.gov/groups/SNS/acpt/documents/ACPT_Concise.pdf)”, Introduction of Access Control Policy Tool in Power Point slides. [http://csrc.nist.gov/groups/SNS/acpt/documents/ACPT\_Concise.pdf]

V. Hu, D.R. Kuhn, T. Xie, J. Hwang, “Model Checking for Verification of Mandatory Access Control Models and Properties”, Int'l Journal of Software Engineering and Knowledge Engineering (IJSEKE).

V. Hu, D.R. Kuhn, T. Xie, "[Property Verification for Generic Access Control Models](http://csrc.nist.gov/groups/SNS/acts/documents/hu-kuhn-xie-08.pdf)", IEEE/IFIP International Symposium on Trust, Security, and Privacy for Pervasive Applications, Shanghai, China, Dec. 17-20, 2008. [http://csrc.nist.gov/groups/SNS/acts/documents/hu-kuhn-xie-08.pdf]

J. Hwang, T. Xie, V. Hu “[Detection of Multiple-Duty-Related Security Leakage in Access Control Policies](http://csrc.nist.gov/groups/SNS/acpt/documents/jhwang4_ssiri09_final.pdf)”, The Third IEEE international conference on Secure Software Integration and reliability improvement (SSIRI), Beijing China, July 8-10, 2009. [http://csrc.nist.gov/groups/SNS/acpt/documents/jhwang4\_ssiri09\_final.pdf]

E. Martin, J. Hwang, T. Xie, and V. Hu, “[Assessing Quality of Policy Properties in Verification of Access Control Policies](http://csrc.nist.gov/groups/SNS/acpt/documents/draft3.pdf)”, Annual Computer Security Applications Conference (ACSAC), December, 2008.Anaheim, California. [http://csrc.nist.gov/groups/SNS/acpt/documents/draft3.pdf]

V. Hu, E. Martin, T. Xie, “[Conformance Checking of Access Control Policies Specified in XACML](http://csrc.nist.gov/groups/SNS/acpt/documents/ac-testing3.pdf)”, proceeding The First IEEE International Workshop on Security in Software Engineering (IWWSE), Beijing China, July 23-27, 2007. [http://csrc.nist.gov/groups/SNS/acpt/documents/ac-testing3.pdf]

[NIST/NSA Privilege (Access) Management Workshop](http://csrc.nist.gov/news_events/privilege-management-workshop/). [http://csrc.nist.gov/news\_events/privilege-management-workshop/]

[Testing and Analysis of Security Policies](http://www.nist.gov/cgi-bin/exit_nist.cgi?url=http://people.engr.ncsu.edu/txie/research.htm#testpolicy) Web site from Computer Science Department of North Carolina State University. [http://www.nist.gov/cgi-bin/exit\_nist.cgi?url=http://people.engr.ncsu.edu/txie/research.htm#testpolicy]