



## **Section Two: Analytics, Policy, Risk and Certifications**

**IdentityIQ Implementation** Training for SailPoint

IdentityIQ Version 6.2

11305 Four Points Drive  
Bldg 2, Suite 100  
Austin, TX 78726

[www.sailpoint.com](http://www.sailpoint.com)

## Contents

Section 2: Analytics, Policies, Risk and Certifications .....	4
Exercise #1: Handling Uncorrelated Identities and Accounts .....	5
Identify and deal with Uncorrelated Accounts.....	5
Exercise #2: Configuring Account Attributes .....	8
Configure Account Attribute Mappings .....	8
Configure the UI .....	12
Investigate the Data .....	14
Exercise #3: Groups and Populations .....	15
Objective.....	15
Overview .....	15
Using Group Factories to Generate Groups .....	15
Generate Populations.....	18
Use Groups and Populations .....	21
Exercise #4: Create Policies .....	22
Objective.....	22
Overview .....	22
Create an Entitlement Separation of Duties Policy.....	22
Create a Policy to detect more than one account per application .....	24
Create an Advanced rule-based Policy to detect dormant accounts .....	24
Scan Identities for Policy Violations .....	25
Exercise #5: Defining Identity Risk Scoring.....	27
Objective:.....	27
Overview: .....	27
Define Identity Risk Model .....	27
Compute Identity Risk Scores.....	29
Exercise #6: Certification of PAM Application and Account Groups.....	30
Objective.....	30
Overview .....	30
Generate an Application Owner Certification.....	30
Perform the Certification as Patrick Jenkins .....	32

Create an Account Group Certification .....	35
Perform the Account Group Certification.....	35
Exercise #7: Manager Certification with Rules.....	36
Objective.....	36
Overview .....	36
Create a Certification for Managers using an Exclusion Rule.....	36
Create a Certification for Managers using a Pre-Delegation Rule .....	38

## Section 2: Analytics, Policies, Risk and Certifications

Now that we have finished onboarding applications, we need to start to use this aggregated data to perform a number of activities including the following:

- Detect and handle Uncorrelated Accounts
- Configure Account Attributes
- Configure Groups and Populations
- Define Policies to detect issues with a user's access
- Define Risk scoring to identify risky users
- Perform Certifications on the following:
  - Application Accounts
  - Application Entitlements
  - Application Account Groups
  - Advanced Certifications based on Populations and Groups
  - Using rules to control the certification behavior:
    - Exclusion Rule
    - Pre-Delegation Rule

## Exercise #1: Handling Uncorrelated Identities and Accounts

### Identify and deal with Uncorrelated Accounts

Earlier, we configured the system to show uncorrelated accounts in the UI by adding the “Authoritative?” column to the UI. This column shows us all Identity Cubes that were created based on a failed correlation from a non-authoritative source. Each of these cubes is housing one or more uncorrelated accounts.

1. Go to the Identities view and sort on the Authoritative column until you see all the uncorrelated Identities at the top.

#### Identities

Filter by Identity Name										
User Name	First Name	Last Name	Manager	Assigned Role Sur	Detected Role Sur	Risk Score	Last Refresh	Status	Authoritative?	
spadmin	The	Administrator				0	12/30/13 5:30 PM		false	
AngieBell						250	12/31/13 2:39 PM		false	
JeffMurphy						250	12/31/13 2:39 PM		false	
FloJohnston						250	12/31/13 2:39 PM		false	
WendyGeorge						250	12/31/13 2:39 PM		false	
John Conner						0	12/31/13 1:33 PM		false	
PRISM ADMIN						0	12/31/13 1:56 PM		false	
Thomas.Martinez	Thomas	Martinez	Robert.Brown			0	12/31/13 2:04 PM	Contractor	true	

2. Click **AngieBell** and notice that she is housing an account on the **Financials** application:

#### View Identity AngieBell

Attributes	Entitlements	Application Accounts	Policy	History	Risk	Activity	User Rights	Events
<b>Application Accounts</b>								
Application			Account Name			Status		
<input type="checkbox"/> Financials			AngieBell			Active		

3. Search for **Angela.Bell** and notice that she is missing an account on the **Financials** application:

## View Identity Angela.Bell

Attributes	Entitlements	Application Accounts	Policy	History	Risk	Activity	User Rights	Events
<b>Application Accounts</b>								
Application	Account Name							
<input type="checkbox"/> EnterpriseApps - AuditReports ▼	Angela Bell							
<input type="checkbox"/> HR System - Employees ▼	Angela.Bell							
<input type="checkbox"/> LDAP ▼	Angela.Bell							
<input type="checkbox"/> Logical Application – TRAKK ▼	Angela.Bell							
<input type="checkbox"/> TRAKK ▼	Angela.Bell							
<input type="button" value="Delete"/> <input type="button" value="Move Account"/>								

- When correlation fails, we get uncorrelated identities created to house the accounts that did not correlate. There are two solutions to this problem:
  - Figure out what went wrong with the correlation, adjust your correlation rule and re-aggregate the accounts.
  - Manually correlate the accounts using the UI. This involves moving the uncorrelated account to the proper identity.

Once this is done, you can prune the Identity cubes that have no accounts to remove the uncorrelated cubes.

- In order to manually correlate accounts, navigate to **Manage→ Identity Correlation** and select the **Financials** application
- Select **AngieBell** from the list of uncorrelated accounts

## Identity Correlation

Use the following tables to manually correlate one or more accounts with an identity. To begin, enter an application name to re columns in the tables can be edited.

Select Uncorrelated Accounts	
Financials ▼	Account ID or Name 🔍 <input type="button" value="Included Account Types ▼"/>
Account ID	Account Name ▲
<input type="checkbox"/> <a href="#">337</a>	AngieBell
<input type="checkbox"/> <a href="#">339</a>	FloJohnston
<input type="checkbox"/> <a href="#">338</a>	JeffMurphy
<input type="checkbox"/> <a href="#">341</a>	WendyGeorge

- In the section **Select Target Identity** search for **Angela** in order to find the appropriate Identity to house this account:

**Select Target Identity**

Angela

Name	First Name	Last Name	Correlated
<input checked="" type="checkbox"/> Angela.Bell	Angela	Bell	<input checked="" type="checkbox"/>

8. Select **Perform Merge** on the bottom right of the screen. This will move the uncorrelated account from the uncorrelated cube to the correlated cube.
9. After the merge is complete, navigate to **Angela.Bell** and **AngieBell** and confirm that the account was moved properly. The **Financials** account should be on the proper cube.

### View Identity Angela.Bell

Attributes	Entitlements	Application Accounts	Policy	History	Risk	Activity	User
<b>Application Accounts</b>							
Application			Account Name				
<input type="checkbox"/>	EnterpriseApps - AuditReports	▼	Angela Bell				
<input type="checkbox"/>	Financials	▼	AngieBell				

### View Identity AngieBell

Attributes	Entitlements	Application Accounts
No Application Accounts		

10. Run the task: **Prune Identity Cubes** and observe the **Task Results** to see that the **AngieBell** Identity cube has been pruned from the environment.

Prune Identity Cubes Attributes	
Attribute	Value
Identities analyzed	236
Identities deleted	1
Identities protected	235
Identities being certified	0
Deletion failures	0

## Exercise #2: Configuring Account Attributes

Sometimes, there are attributes, which we want to be common across all accounts. Common uses for this technique are to store whether an account is inactive, privileged or a system account.

It is common practice to use this technique to “normalize” attributes like last login date, privileged account status, active/inactive status, etc. This allows us to treat these attributes consistently across all accounts.

For our different account attributes, we will build them as follows:

**Privileged** attribute will be sourced from

- **app2\_privileged** attribute from **Financials** application
- **groups** (if it contains Super) value from the **PRISM** application
- **Permission Group** (if it contains ADMINISTRATORS) value from the **PAM** application

**Service** attribute will be sourced from

- **app2\_service** attribute from the **Financials** application

**Inactive** attribute will be sourced from

- **app2\_inactive** attribute from the **Financials** application
- **status** attribute from the **PRISM** application (“A” indicates Active.)

### ***Configure Account Attribute Mappings***

1. Configure Account Attributes for privileged, service and system accounts
  - a. Navigate to **System Setup → Account Mappings**
  - b. Select **Add New Attribute**
  - c. Configure the new attribute, privileged:
    - i. Attribute Name: **privileged**
    - ii. Display Name: **Privileged Account**
    - iii. Attribute Type: **boolean**
    - iv. Click **Add Source**
      1. Choose **Application Attribute**
      2. Application: **Financials**



3. Attribute: **app2\_privileged**
  4. Click **Add**
- v. Click **Add Source**
1. Choose **Application Rule**
  2. Application: **PRISM**
  3. Create a new rule by clicking on the ...
    - a. Rule Name: **Link Attribute – PRISM Privileged**
    - b. Script: Copy and Paste from  
**/home/spadmin/ImplementerTraining/beanshell/Link  
Attribute-PRISM-Privileged.txt**
  4. Click **save** (for Rule Editor)
  5. Make sure to choose the rule you just configured
  6. Click **Add**
- vi. Click **Add Source**
1. Choose **Application Rule**
  2. Application: **PAM**
  3. Create a new rule by clicking on the ...
    - a. Rule Name: **Link Attribute – PAM Privileged**
    - b. Script: Copy and Paste from  
**/home/spadmin/ImplementerTraining/beanshell/Link  
Attribute-PAM-Privileged.txt**
  4. Click **save**
  5. Make sure to choose the rule you just configured
  6. Click **Add**

vii. Your mappings should look like this:

Source Mappings	
1. app2_privileged from the Financials application	
2. Application rule Link Attribute – PRISM Privileged for the PRISM application	^
3. Application rule Link Attribute – PAM Privileged for the PAM application	v

viii. Click **Save**

d. Select **Add New Attribute**

e. Configure the new attribute, service:

- i. Attribute Name: **service**
- ii. Display Name: **Service Account**
- iii. Attribute Type: **boolean**
- iv. Click **Add Source**
  1. Choose **Application Attribute**
  2. Application: **Financials**
  3. Attribute: **app2\_service**
  4. Click **Add**
- v. Click **Save**

f. Select **Add New Attribute**

g. Configure the new attribute, inactive:

- i. Attribute Name: **inactive**
- ii. Display Name: **Inactive Account**
- iii. Attribute Type: **boolean**
- iv. Click **Add Source**
  1. Choose **Application Attribute**
  2. Application: **Financials**
  3. Attribute: **app2\_inactive**

4. Click **Add**
- v. Click **Add Source**
  1. Choose **Application Rule**
  2. Application: **PRISM**
  3. Create a new rule by clicking the ...
    - a. Rule Name: **Link Attribute – PRISM Inactive**
    - b. Script: Copy and Paste from  
**/home/spadmin/ImplementerTraining/beanshell/Link  
 Attribute-PRISM-Inactive.txt**
  4. Click **save**
  5. Make sure to choose the rule you just configured
  6. Click **Add**
  7. Your mappings should look like this:

Source Mappings	
1. app2_inactive from the Financials application	^
2. Application rule Link Attribute – PRISM Inactive for the PRISM application	v

- vi. Click **Save**
- h. Once complete, your **Account Attributes** should look like this:

### Account Attributes

Attribute ^	Primary Source Mapping
Inactive Account	app2_inactive from the Financials application
Privileged Account	app2_privileged from the Financials application
Service Account	app2_service from the Financials application

Page 1 of 1

[Add New Attribute](#)
[Return to System Setup](#)

## Configure the UI

The next goal is to configure the UI to leverage the newly created account attributes. In this section we will configure the UI to display icons indicating the account status of a user. The icon will indicate at a glance any accounts with one of the three attributes previously added: inactive, privileged, and service.

- Under **System Setup → Import from File**, select **Browse...** and import:  
**/home/spadmin/ImplementerTraining/config/AccountIconConfig.xml**
- Execute the tasks:
  - **Aggregate Financial Application**
  - **Aggregate PRISM**
  - **Aggregate PAM**
- Navigate to **Define → Applications → Financials → Accounts**. Notice the new icons, and expand any account and confirm the new account attributes:

Attributes	Schema	Correlation	Accounts	Risk	Activity Data
Filter by Name <input type="text"/>					
Account ID	Account Name				
112	RichardJackson				
113	MariaWhite				
114	CharlesHarris				
<div> <div>Inactive Account</div> <div>Privileged Account</div> <div>Service Account</div> </div>					
acct_lastLogin 09/17/2012 21:26:43					

- Navigate to **Define → Identities** and check the **Financials** and **PRISM** accounts for the following users and confirm that the Privileged, Service, and Inactive indicators are working:

- Adam.Kennedy**

[View Identity Adam.Kennedy](#)

Attributes	Entitlements	Application Accounts	Policy	History	Risk	Activity	User Rights	Events
Application Accounts								
Application	Account Name							
<input type="checkbox"/> EnterpriseApps - CRMPortal	Adam Kennedy							
<input type="checkbox"/> Financials	AdamKennedy							

- Bonnie.Carroll**

## View Identity Bonnie.Carroll

Attributes	Entitlements	Application Accounts	Policy	History	Risk	Activity	User Rights	Events
Application Accounts								
Application			Account Name					
<input type="checkbox"/>	Contractor Feed	▼	Bonnie.Carroll					
<input type="checkbox"/>	Financials	▼	BonnieCarroll					

## c. Walter.Henderson

## View Identity Walter.Henderson

Attributes	Entitlements	Application Accounts	Policy	History	Risk	Activity	User Rights	Events
Application Accounts								
Application			Account Name				Status	
<input type="checkbox"/>	EnterpriseApps - AccountPay	▼	Walter Henderson				Active	
<input type="checkbox"/>	HR System - Employees	▼	Walter.Henderson				Active	
<input type="checkbox"/>	LDAP	▼	Walter.Henderson				Active	
<input type="checkbox"/>	Logical Application – TRAKK	▼	Walter.Henderson				Active	
<input type="checkbox"/>	PRISM	▼	whenderson				Locked	
<input type="checkbox"/>	TRAKK	▼	Walter.Henderson				Active	

**Note:** If you are missing icons: 1) check that the attribute mapping is correct. 2) check for mismatches between the attribute name (for example **P**rivileged) and the UIConfig Object (for example **p**rivileged).

## Investigate the Data

1. Since we added these account attributes as searchable, we can also use analytics to mine identities based on these attributes in **Analyze → Advanced Analytics**

### Advanced Analytics

The screenshot shows the 'Advanced Search' interface. At the top, there are tabs for 'Identity Search', 'Access Review Search', 'Role Search', 'Account Group Search', 'Activity Search', 'Audit Search', and 'Pro'. Below these is an 'Advanced Search' button. The 'Search Criteria' section is expanded, showing 'Identity Attributes'. Under 'Standard Attributes', there are input fields for Last Name, First Name, Username, Display Name, Email, Manager, Is Inactive, and Is Manager. Under 'Searchable Attributes', there are dropdown menus for Location, Employee ID, Region, Status, Privileged Account, Service Account, and Inactive Account. The 'Privileged Account', 'Service Account', and 'Inactive Account' dropdowns are highlighted with a red rectangle.

2. Use Advanced Analytics to explore users that have Privileged, Inactive and Service Accounts.
3. When you created the account attributes Privileged, Service, and Inactive, one of the applications you mapped these attribute values from was the Financials application. You mapped the values app2\_inactive, app2\_privileged, and app2\_service.
  - a. Where are app2\_inactive, app2\_privileged, and app2\_service defined?  
\_\_\_\_\_
  - b. How are app2\_inactive, app2\_privileged, and app2\_service populated?  
\_\_\_\_\_

### Bonus Question

1. When mapping from the PRISM and the PAM applications, why did we use rules to specify inactive and privileged values rather than a direct mapping?  
\_\_\_\_\_  
\_\_\_\_\_

## Exercise #3: Groups and Populations

### **Objective**

The objective of this exercise is to use built-in features of the product to organize identities through the use of Groups and Populations. We will also explore some reports that are useful when dealing with Identities.

### **Overview**

For our implementation, we have been asked to generate some groups based on the following Identity Attributes:

- Status (whether an Employee or Contractor)
- Location (Austin, Tokyo, etc.)
- Manager

Because we defined these Identity Attributes as group factories earlier when we defined the identity mappings, it is extremely easy to use IdentityIQ to automatically calculate and generate groups of identities based on these fields. These groups can be used in reporting and other portions of the product.

We will also be using rules to assign ownership to each group.

Additionally, we want to use Advanced Analytics to define some populations based on specific criteria. Populations are similar to groups, except that they are driven off of multiple search criteria whereas Groups are statically defined based off a single Identity attribute.

For our implementation, we want to generate two populations.

- Active Managers who are not Contractors in Asia-Pacific Region only
- All users who have Privileged accounts on any application

### **Using Group Factories to Generate Groups**

We will now configure and generate groups for the Status, Location and Manager attributes on our identities.

1. Load Rules to determine Group ownership
  - a. Navigate to **System Setup → Import from File** and load the following two rules:

**/home/spadmin/ImplementerTraining/config/Rule-GroupOwner-AssignManagerAsOwner.xml**

**/home/spadmin/ImplementerTraining/config/Rule-GroupOwner-HighestRanking.xml**

2. Review the attributes for which the Group Factory option was selected.
  - a. Navigate to **System Setup → Identity Mappings** and list the 6 attributes for which Group Factory was selected:

_____	_____
_____	_____
_____	_____

3. Navigate to **Define → Groups** and select **Create New Group** and fill in the following fields:

### Group Configuration

- a. Name: **Status**
- b. Group Attribute: **Status**
  - i. Notice that the choices for Group Attribute is populated from the list of attributes for which Group Factory was selected.
- c. Description: **Group used to define Employees and Contractors**
- d. Enabled: **Checked**
- e. Group Owner Rule: **Group Owner – Highest Ranking Member of Sub-Group**

### Edit Group

- f. Select **Save**



4. Repeat the same steps for the Location attribute
  - a. Select **Create New Group**
  - b. Name: **Location**
  - c. Group Attribute: **Location**
  - d. Description: **Groups based off of each Identity's location attribute.**
  - e. Enabled: **Checked**
  - f. Group Owner Rule: **Group Owner – Highest Ranking Member of Sub-Group**
  - g. Select **Save**
5. Repeat the same steps for the Manager attribute
  - a. Select **Create New Group**
  - b. Name: **Manager**
  - c. Group Attribute: **Manager**
  - d. Description: **Users grouped by Manager.**
  - e. Enabled: **Checked**
  - f. Group Owner Rule: **Group Owner – Assign Manager**
  - g. Select **Save**

### Group Configuration

Groups Populations Workgroups		
Filter by Group Name		<input type="text"/> <input type="button" value="Create New Group"/>
Name	Attribute	Description
Location	location	Group based off of each Identity's location attribute.
Manager	manager	Group used to group users by Manager.
Status	status	Group used to define Employees and Contractors

6. Generate Groups using the newly created group configurations and confirm that they were created correctly:
  - a. Run the task: **Refresh Groups**

- b. Navigate to **Define → Groups** and check all three group factories to determine if the groups were created correctly. Look for there to be many subgroups and the owner fields should be populated.

Here is an example of the Location groups:

**Sub-Groups**

Name	Member Count	Policy Violations	Composite Score	Owner
Austin	27	0	● 166	James.Smith
Brazil	25	0	● 170	John.Williams
Brussels	26	0	● 173	Jerry.Bennett
London	25	0	● 250	Amanda.Ross
Munich	25	0	● 170	Dennis.Barnes
No Location	6	0	● 208	The Administrator

**Note:** These group themselves are not dynamic. You must run the **Refresh Groups** task periodically to update them. Between runs of **Refresh Groups**, the groups themselves remain static, but the membership is always based off a dynamic query.

### **Generate Populations**

Next we will generate some populations of users that represent some interesting sets of users. Populations can be generated off any of the data that is available via the Advanced Analytics feature of IdentityIQ.

For our implementation, we want to generate two populations.

- Active Managers who are not Contractors in Asia-Pacific Region only
- All users who have Privileged accounts on any application

1. Navigate to **Analyze → Advanced Analytics**
2. Under the **Identity Search** tab, click **Clear Search** and enter the following search criteria:
  - a. Is Inactive: **false**
  - b. Is Manager: **true**
  - c. Region: **Asia-Pacific**
  - d. Status: **Employee**

- e. Click **Run Search**

### Advanced Analytics

The screenshot shows the 'Advanced Analytics' search interface. At the top, there are tabs for 'Identity Search', 'Access Review Search', 'Role Search', 'Account Group Search', 'Activity Search', 'Audit Search', and 'Process Search'. Below these is an 'Advanced Search' button. The 'Search Criteria' section is expanded, showing 'Identity Attributes' and 'Searchable Attributes'. The 'Identity Attributes' section includes fields for Last Name, First Name, Username, Display Name, Email, Manager, Is Inactive (set to False), and Is Manager (set to True). The 'Searchable Attributes' section includes Location, Employee ID, Region (set to Asia-Pacific), Status (set to Employee), Privileged Account, Service Account, and Inactive Account. At the bottom, there is a 'Run Search' button (highlighted with a red box) and a 'Clear Search' button.

- f. You should get 16 results returned.
- g. From the drop down menu, select **Save Identities as Population**

### Advanced Analytics

The screenshot shows the 'Advanced Analytics' search results interface. At the top, there are tabs for 'Identity Search', 'Access Review Search', 'Role Search', 'Account Group Search', and 'Activity Search'. Below these is a 'Search Results - 16 Results Returned' section. The 'Result Options' dropdown menu is open, showing options like 'Save Search', 'Save Search as Report', 'Save Identities as Population' (highlighted with a red box), and 'Show Entitlements'. Below the dropdown, there is a list of search results with checkboxes next to names: Bobby.Stephens, Carlos.Perkins, Eugene.Hawkins, and Howard.Rose.

- h. Name: **Active Managers – Asia-Pacific**
- i. Click **Save**

3. Create another Population with the following criteria
  - a. First click **Refine Search**, then select **Clear Search** to reset everything
  - b. Privileged Account: **True**
  - c. Click **Run Search**
  - d. You should see results showing all users with Privileged accounts
  - e. Save as a Population with the following name: **Identities with Privileged Accounts**
4. Navigate to **Define → Groups** and select the **Populations** tab
  - a. Confirm that you have two populations defined:

#### Group Configuration

Groups Populations Workgroups	
Filter by Population Name <input type="text"/>	
Name	Description
Active Managers – Asia-Pacific	
Identities with Privileged Accounts	

- b. Click either population and see that you can list the members within the given population.

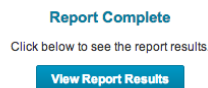
**Note:** By default, these populations are only visible to the user who created them. You can edit the populations and make them Public.

**Note:** Populations are dynamic queries, so every time you view a population, you are viewing its current members at that point in time.

## ***Use Groups and Populations***

Groups and Populations can be used in a number of places in the product. One of those is in reporting. Here we will run a report using a Group to narrow down the results.

1. Generate a report using a Group Factory
  - a. Navigate to **Analyze → Reports** and select the **Reports** tab
  - b. Create a new report of type: **User Details Report**
    - i. On the Standard Properties page:
      1. Report Name: **User Details - Financial App - London**
    - ii. On the Additional Identity Properties page:
      1. Application: **Financials**
      2. Groups: **London**
  - c. Select **Save and Execute**
  - d. Wait for the report to finish and then select: **View Report Results** to see the report results.



2. Generate a report using a Population
  - a. Create a new report of type: **Identity Effective Access Live Report**
  - b. On the Standard Properties page:
    - i. Report Name: **Effective Access – Privileged Users**
  - c. On the Additional Identity Properties page:
    - i. Groups: **Identities with Privileged Accounts**
  - d. Select **Save and Execute**
  - e. Wait for the report to finish and then select: **View Report Results** to see the report results.
3. We have now created two reports (one using a group based off of a group factory and the other using a population.) Later we will see other uses for **Groups** and **Populations**.

## Exercise #4: Create Policies

### Objective

The objective of this exercise is to create some policies. These policies will analyze identity data to determine who has violated the policies we define and to allow managers and other users to learn about the policy violations.

### Overview

Now that we have loaded a rich assortment of account and account group data, we can start to mine this data to determine if we have any Policy Violations in the data set.

The client has requested that we implement three policies.

- No user can simultaneously have the **super** and **input** access to the **TRAKK** application
- No user can have more than one account on any system
- For the **PAM** application, any user who has not used the system in 180 days will be considered in violation of policy

### Create an Entitlement Separation of Duties Policy

1. Navigate to **Define → Policies**
2. In the upper right, click **New Policy** and select **Entitlement SOD Policy**
3. Configure as follows:
  - a. Name: **TRAKK SOD Policy**
  - b. Owner: **The Administrator**
  - c. Violation Owner: **Manager is Violation Owner**
  - d. State: **Active**
  - e. Send Alerts: **Checked**
    - i. Initial Notification Email: **Policy Violation**
    - ii. Observers: **Aaron.Nichols**
  - f. SOD Policy Rules: click **Create New Rule**
    - i. Summary: **Cannot be Super and Input at the same time**
    - ii. First Entitlement Set:
      1. Application Items: **TRAKK**

2. Select **Add Attribute**
3. Select Name: **capability**
4. Select Value: **super**

**First Entitlement Set**

IdentityIQ Items

Add Identity Attribute

---

**Application Items**

TRAKK ▼ Add Attribute Add Permission

---

Operation	Type	Application	Name	Value
Or ▾	<input type="checkbox"/> Attribute	TRAKK	capability ▾	super

Group Selected Ungroup Selected Delete Selected

iii. Second Entitlement Set:

1. Application Items: **TRAKK**
2. Select **Add Attribute**
3. Select Name: **capability**
4. Select Value: **input**

**Second Entitlement Set**

IdentityIQ Items

Add Identity Attribute

---

**Application Items**

TRAKK ▼ Add Attribute Add Permission

---

Operation	Type	Application	Name	Value
Or ▾	<input type="checkbox"/> Attribute	TRAKK	capability ▾	input

Group Selected Ungroup Selected Delete Selected

iv. Click **Done** to complete the rule

**SOD Policy Rules**

Rule ▲	Any of these entitlements...	...conflict with any of these entitlements
Cannot be Super and ...	(capability = "super")	(capability = "input")

- g. Scroll down and click **Save** to complete the policy

***Create a Policy to detect more than one account per application***

1. Navigate to **Define → Policies**
2. In the upper right, click **New Policy** and select **Account Policy**
3. Configure as follows:
  - a. Name: **More than one account**
  - b. Owner: **The Administrator**
  - c. Violation Owner: **Manager is Violation Owner**
  - d. State: **Active**
  - e. Send Alerts: **Checked**
  - f. Initial Notification Email: **Policy Violation**
  - g. Observers: **Aaron.Nichols**
  - h. Summary: **Multiple Application Accounts**
  - i. Scroll down and click **Save** to complete the policy

***Create an Advanced rule-based Policy to detect dormant accounts***

1. Navigate to **Define → Policies**
2. In the upper right, click **New Policy** and select **Advanced Policy**
3. Configure as follows:
  - a. Name: **Last Login more than 180 days ago**
  - b. Owner: **The Administrator**
  - c. Violation Owner: **Manager is Violation Owner**
  - d. State: **Active**
  - e. Send Alerts: **Checked**
  - f. Initial Notification Email: **Policy Violation**
  - g. Observers: **Aaron.Nichols**
  - h. Policy Rules: click **Create New Rule**
    - i. Summary: **Last Login > 180 Days**



- ii. Selection Method: Choose **Rule**
- iii. Click “...” to edit the rule
- iv. Rule Name: **Violation Rule - No login for last 180 days**
- v. Rule body: Copy and paste from the:  
  

**/home/spadmin/ImplementerTraining/beanshell/PolicyViolation-NoLoginFor180Days.txt**
- vi. Click **Save** to save the rule
- vii. Make sure to choose the rule once you’ve saved it, then click **Done**
- i. Click **Save** to save the policy

### ***Scan Identities for Policy Violations***

1. Run the task: **Check Active Policies**

**Note:** This task is an Identity Refresh task with **Check active policies** checked

2. Check the **Task Results** tab when the task ends and confirm:

Check Active Policies Attributes	
Attribute	Value
Identities examined	235
Policies checked	Last Login more than 180 days ago, More than one account, TRAKK SOD Policy
Policy violations	56
Policy notifications	15

3. Confirm that Policy Violations were found. There are several ways you can see policy violations:
  - a. Navigate to **Manage → Policy Violations**. This will show all policy violations. Click any of the TRAKK SOD policy violations to interact with it.
    - i. For **Violation Decision**, choose **Correct Violation** from the dropdown

- ii. Notice that you are presented with an option to remove one of the offending entitlements.

**Violation Decision** Correct Violation ▾

**Correction Advice**  
Select the entitlement(s) that should be revoked to correct this violation.

**Conflicting Entitlements**

**Revoke at least one of the following entitlements**

☐ TRAKK capability super ?

☐ TRAKK capability input ?

- b. Navigate to **Carl.Foster's** cube and check the **Policy** tab to see a policy violation.
- c. Look in each manager's Inbox for incoming workitems for each policy violation detected.
  - i. Check the Administrator's Inbox.  
**Note:** The administrator will get any violations for users who don't have managers
  - ii. Log out and back in as **Aaron.Nichols/xyzzy** and check his Inbox
- d. Check the Email log
  - i. Launch the **Tail Email Log** shortcut and confirm that you can see emails that were sent out when policy violations were discovered.

## Exercise #5: Defining Identity Risk Scoring

### **Objective:**

The objective of this section is to learn how to configure the IdentityIQ risk scoring and apply the configured scoring settings to existing Identities.

### **Overview:**

Our client has stated that they want their risk score to be calculated based on several qualities:

- 50% of the risk score needs to be based on certification age (how recently has the identity been certified)
- 25% of the risk score needs to be based on Policy Violations
  - Having multiple accounts is higher risk
  - Having conflicting role is higher risk
  - Having not logged into PAM for 180 days is lower risk
- 25% of the risk score needs to be based on Entitlements owned by a user
  - Having Super User (super) access to TRAKK is higher risk
  - Having Manager (approve, reject) access to TRAKK is medium risk

### **Define Identity Risk Model**

1. Login as **spadmin/admin**
2. Navigate to **Define → Identity Risk Model**
3. Click the **Composite Scoring** tab
4. Configure:
  - a. Role Compensated Score: 0%
  - b. Entitlement Compensated Score: 25%
  - c. Policy Violation Compensated Score: 25%
  - d. Certification Age: 50%
5. Click the **Baseline Access Risk** tab
  - a. Click **Entitlement Baseline Access Risk** Configuration

- b. When prompted, **Save**
- c. Select **TRAKK** as the application and click **Add**
- d. Choose **Configure Attributes**
- e. Configure the attributes:
  - i. capability: super: **500**
  - ii. capability: approve: **250**

#### TRAKK Attributes

<input type="checkbox"/>	Attribute	Value	Weight
<input type="checkbox"/>	capability	approve	<input type="range"/> 250
<input type="checkbox"/>	capability	super	<input type="range"/> 500

- iii. Click **Save** twice to save the configuration.
- f. Click **Policy Violation Baseline Access Risk** and configure:
  - i. Cannot be Super and Input at the same time: **300**
  - ii. Multiple Application Accounts: **300**
  - iii. Last Login > 180 Days: **100**
  - iv. Click **Save** and then **Yes** to confirm

Rule	Risk Level
Cannot be Super and Input at the same time	<input type="range"/> 300
<b>More than one account Violation</b>	
Multiple Application Accounts	<input type="range"/> 300
<b>Last Login more than 180 days ago Violation</b>	
Last Login > 180 Days	<input type="range"/> 100

## Compute Identity Risk Scores

1. Run the task: **Refresh Risk Scores**
2. Confirm the results once the task is done running:

Refresh Risk Scores Attributes	
Attribute	Value
Identities examined	235
Scores changed	186

3. Confirm the scoring in several places.
  - a. Navigate to any identity cube and check the **Risk** tab to see if risk scoring has been updated. Here is **Richard.Jackson's** cube:

Attributes

Entitlements

Application Accounts

Policy

History

Risk

Activity

User Rights

Events

Scorecard

Score Category	Base Score	Compensated Score
Role Compensated Score	<div><div></div>0</div>	<div><div></div>0</div>
Entitlement Compensated Score	<div><div></div>760</div>	<div><div></div>760</div>
Policy Violation Compensated Score	<div><div></div>600</div>	<div><div></div>600</div>
Certification Age	<div><div></div>1000</div>	<div><div></div></div>

Top Composite Score Contributors

Score Category	Contributor	Score	Percentage of Total
Certification	Identity has not been certified	1000	60%
Entitlement	TRAKK : capability = Input,reject,approve,super	752	22%
Policy	TRAKK SOD Policy : Cannot be Super and Input at the same time	300	9%
Policy	More than one account : Multiple Application Accounts	300	9%

- b. Navigate to **Manage → Identity Risk Scores**
  - i. This view organizes Risk scores into Low/Medium/High groupings
  - ii. A user can schedule certifications for identities from this screen
- c. Use Advanced Analytics to search based on Risk Score criteria. **Note:** You could use risk scoring as part of the criteria to create populations.

## Exercise #6: Certification of PAM Application and Account Groups

### Objective

Certify the PAM Application and the Account Groups that accompany it

### Overview

Our customer wants us to perform an initial certification of all PAM application accounts and the PAM Account Groups as well. We will do this by kicking off a certification against the Application Owner and the Owner of the Account Groups.

When creating the PAM application, we configured an Application owner for the PAM application: Patrick.Jenkins. The Account Groups on the PAM application are owned by Patrick.Jenkins as well, unless we decide to define individual Account Group owners as well.

### Generate an Application Owner Certification

1. Navigate to **Monitor** → **Certification** and from the drop-down on the right, select **Application Owner** certification.
2. Under the **Basic** section, configure the following:
  - a. Certification Name: **PAM Application Owner Certification [\${fullDate}]**
  - b. Certification Owner: **The Administrator**
  - c. Applications: **PAM**
  - d. Run Now: **Checked**
3. Under the **Lifecycle** section, configure the following:
  - a. Enable Staging Period: **checked**
4. Choose **Schedule Certification**
5. **Note:** It can take awhile for the certification to generate. Eventually it will show up in the **Monitor** → **Certifications** list. During this time, the system is building the certification. You can hit the refresh button periodically until the Certification shows up in a **Staged** status.

Certifications

New Certification

Application certifications scheduled successfully.

Certifications

Certification Schedules

Certification Events

Search by Certification Name

Q

Advanced Search

Name	Owner	Status	Percent Complete	Create Date	Tags
PAM Application Owner Certification [1/2/14 9:38:...	The Administrator	Staged	0% (0 of 1)	01/02/14 03:38:45 pm	

6. Once it shows up, click the certification, to see a staged view of the certification. During staging, the certification owner can review the entire certification, and decide whether to **Activate** or **Cancel Certification**.

#### PAM Application Owner Certification [1/2/14 9:38:45 AM CST]

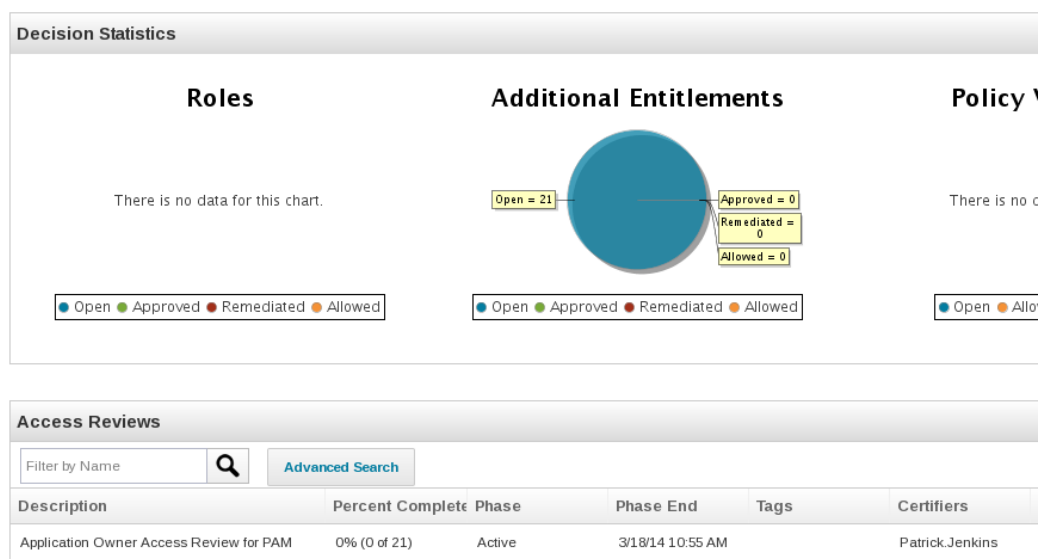
Owner The Administrator  
 Create Date 1/2/14 3:38:45 PM WET  
 Exclusions 0  
[\[View/Edit Certification Options\]](#)

Access Reviews Completed 0/1 (0%)  
 Identities Completed 0/7 (0%)  
 Items Completed 0/14 (0%)



#### Decision Statistics

- Scroll down to the **Access Reviews** section and see that the overall certification consists of one Access Review assigned to Patrick.Jenkins (the Owner of the PAM application.) View the access review details by selecting the **Application Owner Access Review for PAM**
  - At the bottom of the page, click **Back**
7. Click **Activate** to send the certification out to the reviewers. You can always return to the overview page to see the current status of the active certification. At this point, the certification is showing 0% complete, as we would expect.



8. Investigate the certification and answer the following questions:
- How many Access Reviews are included in this certification? \_\_\_\_\_
  - How many identities are included in the first Access Review? \_\_\_\_\_
  - Why is Patrick.Jenkins the certifier? \_\_\_\_\_

## Perform the Certification as Patrick Jenkins

1. Log out and back in as **Patrick.Jenkins/xyzzzy**
2. On the dashboard, you can see the following shortcuts. Clicking on **Access Reviews** will take you to the access review for **Patrick.Jenkins**

### Dashboard

#### COMPLIANCE ACTIVITIES



Access Reviews (1)  
Policy Violations (0)

#### ASSIGNED TASKS



Approvals (0)  
Sign-off Reports (0)  
Work Items (1)

**Note:** You could also go to **Manage → My Access Reviews**, or select the **Work Items** link to see all assigned work items, including access reviews

3. You can choose to perform this access review one of two ways (Worksheet View or Identity View.)
4. Either route you follow, approve everything except the three entitlements for the user **John Connor**. For the three entitlements for **John Connor**, revoke them and select **Save Changes**.

Legend:  Approve  Revoke  Allow Exception  Action Required							
<input checked="" type="checkbox"/>	Decision	Identity	First Name	Last Name	Description	Application	Account Nam
<input type="checkbox"/>		Carl.Foster	Carl	Foster	Value ACCOUNTING on Permission Group	PAM	Carl Foster
<input type="checkbox"/>		Carl.Foster	Carl	Foster	Value TEST01 on Database Name	PAM	Carl Foster
<input type="checkbox"/>		James.Smith	James	Smith	Value ACCOUNTING on Permission Group	PAM	James Smith
<input type="checkbox"/>		James.Smith	James	Smith	Value TEST01 on Database Name	PAM	James Smith
<input type="checkbox"/>		John Conner			Value IT on Permission Group	PAM	John Conner
<input type="checkbox"/>		John Conner			Value ADMINISTRATORS on Permission Group	PAM	John Conner
<input type="checkbox"/>		John Conner			Value TEST01 on Database Name	PAM	John Conner
<input type="checkbox"/>		John.Willia...	John	Williams	Value FINANCE on Permission Group	PAM	John Willia...
<input type="checkbox"/>		John.Willia...	John	Williams	Value HR on Permission Group	PAM	John Willia...
<input type="checkbox"/>		John.Willia...	John	Williams	Value IT on Permission Group	PAM	John Willia...



5. Once done, scroll up and **Sign Off**, then select **Finish**.

### Access Review Details

Application Owner Access Review for PAM			
Due on	2/2/14 (31 Days remaining)	Current Phase	Active (31 Days remaining)
Owner	Patrick.Jenkins	Percent Complete	<div style="width: 100%; background-color: green;"></div> 21/21 (100%)
To complete the access review you must sign off on all decisions.			
<a href="#" style="background-color: #0070C0; color: white; padding: 5px 15px; text-decoration: none;">Sign Off</a>			

6. If we had provisioning configured for this application, we could create an external help ticket or directly de-provision to the target resource. Since we don't have either configured, an IdentityIQ work item will be generated and delivered to **Albert.Woods** (the revoker configured for the PAM application.)
7. Logout and log in as **Albert.Woods/xyzzz**. Look in his dashboard and Inbox, for a Remediation work item. Note it may take awhile, because we must wait for the certification to finish. Notice that a user can click the Work Items link or look in the Inbox to see the remediation work item.

### Dashboard

**COMPLIANCE ACTIVITIES**

 Access Reviews (0)  
Policy Violations (0)

**ASSIGNED TASKS**

 Approvals (0)  
Sign-off Reports (0)  
Work Items (1)

Inbox					
Filter by Item Name or ID				Advanced Search	
Name	Type	Requester	Created	Expiration	Priority
Remediations from 'PAM Access Review' for Albert.Woods	Remediation	Patrick.Jenkins	1/2/14	2/1/14	Normal

Page 1 of 1    Displaying 1 - 1 of 1

8. Click this work item. Notice that this work item contains the remediations that were asked for during the access review:

Work Item ID 60

Requester Patrick.Jenkins

Owner Albert.Woods

Description Remediations from 'PAM Access Review' for Albert.Woods

Created Jan 2, 2014 5:34:47 PM

Next Event Date Feb 1, 2014 5:34:47 PM

Expiration Feb 1, 2014 5:34:47 PM

Priority Normal

History None

Send Comment to Requester

None

[Add Comment](#)

<input type="checkbox"/>	Name	First Name	Last Name	Application	Account	Completed	Entitlements
<input type="checkbox"/>	John Conner			PAM	T1T2T3	N/A	Remove IT from Permission Group for T1T2T3
<input type="checkbox"/>	John Conner			PAM	T1T2T3	N/A	Remove ADMINISTRATORS from Permission Group for T1T2T3
<input type="checkbox"/>	John Conner			PAM	T1T2T3	N/A	Remove TEST01 from Database Name for T1T2T3

9. At this point, Albert could revoke these entitlements on the PAM application manually and mark them as complete. Later in the class we will see how these changes can be provisioned automatically.













## Create an Account Group Certification

Account Group ownership will default to the owner of the application, but you could define account group owners if you wanted to. In our case, we will leave all the account groups without an owner, which means that ownership will default to the PAM application owner for all of the PAM Account Groups. This time around, we will not stage the certification.

1. Logout and log in as **spadmin/admin**
2. Navigate to **Monitor → Certifications** and from the drop down, select **Account Group Permissions**
3. Under the **Basic** Section, configure the following:
  - a. Name: **PAM Account Group Permissions Certification [\${fullDate}]**
  - b. Certification Owner: **The Administrator**
  - c. Applications: **PAM**
  - d. Run Now: **Checked**
4. Click **Schedule Certification**

## Perform the Account Group Certification

1. Log in as **Patrick.Jenkins/xyzzy** and open up the access review
2. Within the access review, click **FINANCE**. Notice that this certification is different from the application owner certification. We are now certifying the individual permissions that make up the Account Group **FINANCE**.

Decisions		Group Information
<div> <span>Approve All</span> <span>Revoke All</span> <span>Delegate All</span> <span>Clear Decisions</span> </div>		
<div>           Legend: <span>Approve</span> <span>Revoke</span> <span>Allow Exception</span> <span>Action Required</span> </div>		
Account Group Permissions		
Decision	Attribute	Entitlements
 	Permission Rights	DR System:YY-Function Control:NN-BackupControl:YY
 	BackupControl	create
 	BackupControl	update
 	DR System	create
 	DR System	update
 	Function Control	execute
<div> <span>Page 1 of 1</span> <span>Show 15 items</span> </div>		

3. Do not perform the certification; just continue to the next exercise.

## Exercise #7: Manager Certification with Rules

### Objective

The objective of this exercise is to certify all the employees within a manager's department but exclude all inactive and contractor identities. We will also run the certification a second time, using pre-delegation rules to assign all inactive identities to the **The Administrator**.

### Overview

In order to accomplish this objective, we will need to kick off a manager certification for a specific manager (in this case we will perform a certification for Catherine Simmons' direct reports.)

Catherine Simmons has five direct reports. Of these, three are Employees, and two are Contractors. One of the identities (Denise Hunt) is currently inactive, as she has left the company. If you want to confirm, use **Advanced Analytics** to search for users with manager **Catherine Simmons**.

Identity Search		Access Review Search	Role Search	Account Gr
Search Results - 5 Results Returned				
Result Options		Refine Search	Schedule Certification	
Username			Status	
<input type="checkbox"/>	Denise.Hunt		Employee	
<input type="checkbox"/>	Irene.Mills		Employee	
<input type="checkbox"/>	Jeremy.Palmer		Contractor	
<input type="checkbox"/>	Louis.Black		Employee	
<input type="checkbox"/>	Tammy.Daniels		Contractor	

We will perform a certification for this department, but we are going to create a special type of rule called an exclusion rule to exclude all contractors and inactive identities. If we set this up correctly, the only users that get certified will be Irene Mills and Louis Black.

In order to perform the pre-delegation, we will use a special rule called a pre-delegation rule to assign the access reviews to a different user (**spadmin**) if an account is inactive.

### Create a Certification for Managers using an Exclusion Rule

1. Logout and log in as **spadmin/admin**
2. Navigate to **Monitor → Certifications** and from the drop down, select **Manager**
3. Under the **Basic** Section, configure the following:
  - a. Name: **Manager Certification – Active Employees [\${fullDate}]**
  - b. Certification Owner: **The Administrator**

- c. Recipient: **Catherine.Simmons**
  - d. Run Now: **Checked**
5. Under the **Advanced** Section, configure the following:
- a. Generate Certification(s): **For the specified managers only**
  - b. Exclusion Rule:
    - i. Click “...”
    - ii. Name: **Exclusion Rule - Manager Cert**
    - iii. Copy and paste the beanshell code from  
**/home/spadmin/ImplementerTraining/beanshell/Manager-Cert-Exclusion-Rule.txt**
    - iv. **Save** the Rule
    - v. Make sure to **Select** the newly created rule once you are done editing it.
6. Click **Schedule Certification**
7. From the desktop, run the shortcut **Tail Tomcat Standard Out** and notice the output messages from the exclusion rule. You can see the logic progression as we walked through all the direct reports to determine who we should certify:

```

Entering Exclusion Rule.
Identity is Inactive : Denise.Hunt
Do not certify.
Entering Exclusion Rule.
Identity is Active and Employee: Irene.Mills
Do the certification.
Entering Exclusion Rule.
Identity is a Contractor: Jeremy.Palmer
Do not certify.
Entering Exclusion Rule.
Identity is Active and Employee: Louis.Black
Do the certification.
Entering Exclusion Rule.
Identity is a Contractor: Tammy.Daniels
Do not certify.

```

8. Also, login as **Catherine.Simmons/xyzzy** and notice that the final Account Review itself is only for two identities:

Legend:  Approve  Revoke  Allow Exception  Action Required					
<input type="checkbox"/>	Decision	Identity	First Nam	Last Name	Description
<input type="checkbox"/>		Irene.Mills	Irene	Mills	Value Treasury on groupmbr
<input type="checkbox"/>		Irene.Mills	Irene	Mills	Value TR-Hedge on groupmbr
<input type="checkbox"/>		Irene.Mills	Irene	Mills	Value Input on capability
<input type="checkbox"/>		Louis.Black	Louis	Black	Value Treasury on groupmbr
<input type="checkbox"/>		Louis.Black	Louis	Black	Value TR-Hedge on groupmbr
<input type="checkbox"/>		Louis.Black	Louis	Black	Value Input on capability

### Create a Certification for Managers using a Pre-Delegation Rule

1. Logout and login as **spadmin/admin**
2. Navigate to **Monitor → Certifications** and from the drop down, select **Manager**
3. Under the **Basic** Section, configure the following:
  - a. Name: **Manager Certification – All Personnel [\${fullDate}]**
  - a. Certification Owner: **The Administrator**
  - b. Recipient: **Catherine.Simmons**
  - c. Run Now: **Checked**
4. Under the **Advanced** Section, configure the following:
  - a. Generate Certification(s): **For the specified managers only**
  - b. Scroll down to the **Certification Rules** section and configure a **Pre-Delegation Rule**:
    - i. Click “...”
    - ii. Name: **Predelegation Rule - Manager Cert**
    - iii. Copy and paste the beanshell code from  
**/home/spadmin/ImplementerTraining/beanshell/Manager-Cert-Pre-Delegation-Rule.txt**
    - iv. **Save** the Rule
    - v. Make sure to **Select** the rule once you are done editing it.

5. Click **Schedule Certification**

6. In the tail program for standard out, notice the output messages from the pre-delegation rule. You can see the logic progression as we walked through all the direct reports to determine whom we should delegate to spadmin (The Administrator):

```

Entering Pre-Delegation Rule
Identity being certified = Denise.Hunt
Identity is Inactive, so pass delegation off to spadmin user.
Entering Pre-Delegation Rule
Identity being certified = Irene.Mills
Identity is Active, so proceed with delegation as usual.
Entering Pre-Delegation Rule
Identity being certified = Jeremy.Palmer
Identity is Active, so proceed with delegation as usual.
Entering Pre-Delegation Rule
Identity being certified = Louis.Black
Identity is Active, so proceed with delegation as usual.
Entering Pre-Delegation Rule
Identity being certified = Tammy.Daniels
Identity is Active, so proceed with delegation as usual.

```

7. Confirm that the Pre-Delegation rule worked

- As **spadmin**, navigate to your inbox and confirm that you have an access review delegated to you (in the form of a work item) for the user: **Denise Hunt**
- Navigate to **Monitor → Certifications** and click **Manager Certification – All Personnel**
- Scroll down under Access Reviews, and select the Access Review: **Manager Access Review for Catherine Simmons**
- Note that all of Denise Hunt's certification items show that they have been delegated.

Legend:  Approve  Revoke  Allow Exception  Action Required							
<input type="checkbox"/> Decision	Identity	First Name	Last Name	Description	Application	Account Name	Status
<input type="checkbox"/>	Denise.Hunt	Denise	Hunt	Value Treasury on groupmbr	Financials	DeniseHunt	Delegated
<input type="checkbox"/>	Denise.Hunt	Denise	Hunt	Value TR-Hedge on groupmbr	Financials	DeniseHunt	Delegated
<input type="checkbox"/>	Denise.Hunt	Denise	Hunt	Value Input on capability	TRAKK	Denise.Hunt	Delegated
<input type="checkbox"/>	Irene.Mills	Irene	Mills	Value Treasury on groupmbr	Financials	IreneMills	Open
<input type="checkbox"/>	Irene.Mills	Irene	Mills	Value TR-Hedge on groupmbr	Financials	IreneMills	Open

- The Access Review pre-delegation was successful because the inactive user was properly delegated to The Administrator.