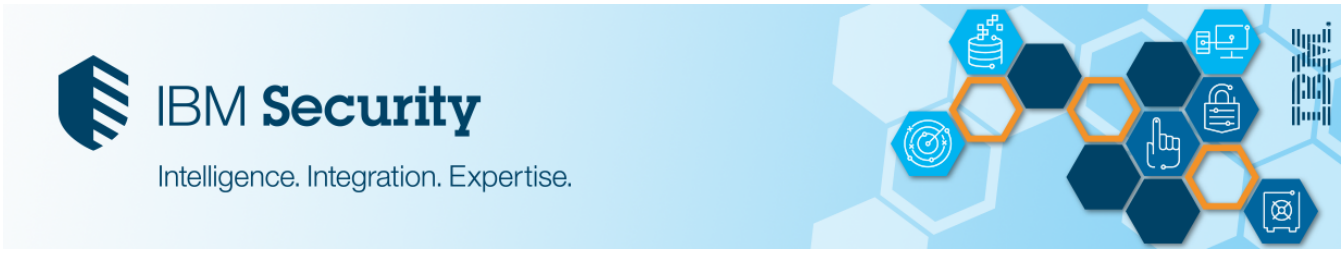



# IBM SECURITY IDENTITY GOVERNANCE AND INTELLIGENCE

## Advanced Scenario Labs (Lab10) for the MasterSkills University 2018

**5.2.x**

**David Edwards**

Version 0.1
April 2018

# Document Purpose

This document provides the instructions for running the labs for the advanced scenarios for the 2018 MasterSkills University – IGI Track.

For any comments/corrections, please contact David Edwards (davidedw@au1.ibm.com).

# Document Conventions

The following conventions are used in this document:

☐ A step to be performed by the student.

A note, some special information or warning.

```
A piece of code
```

Normal paragraph font is used for general information. **Bold** is used to highlight something in an instruction, like a command or menu selection.

The term “IGI” is used to refer to IBM Security Identity Governance and Intelligence.

# Document Control

| Release Date | Version | Authors | Comments |
|---|---|---|---|
| 06 Apr 2018 | 0.1 | David Edwards | Initial version |

# Table of Contents

# 1 Introduction to the Lab

This document is a lab guide for some advanced labs developed for the 2018 Master Skills University – IGI Track.

There are three labs covered in this guide:

1) An end to end AD integration with reconciliation and provisioning, comprising:
   a) AD adapter integration and reconciliation
   b) AD provisioning and attribute policy enforcement
2) A user department move drives user access re-certification

The labs are based on the IGI Virtual Machine (VM) that contains IGI, the Brokerage layer, the IGI database and the Brokerage LDAP.

Where detailed steps are shown for the labs, the steps to be performed are shown by the square (□) beside them.

# 2 Lab Pre-Requisites

This section defines the lab pre-requisites.

## 2.1 Expected Knowledge

This lab assumes the following knowledge has been acquired before attempting the labs:

- Familiarity with IGI, the data objects and concepts, processes and activities, the Admin Console and the Service Center
- These labs require knowledge of, and experience with, advanced configuration in IGI including Java rules. The detailed instruction steps will guide you through configuration of these but knowledge of these would be helpful

This knowledge can be gained via the introductory (Foundation or Basic) training of IGI.

## 2.2 Standard Lab Setup

This lab uses the standard IGI training lab. Setup for this lab is described in the document ***Lab00 - IGI Lab Environment Setup Guide***.

These documents describe the standard training environment used for the IGI labs and the steps to prepare for this lab.

The AD lab requires the Windows server. This is provided as standard with the Skytap templates and SCS-Portal profiles but must be downloaded for the locally run VMs.

## 2.3 Additional Lab Setup

This lab uses a different Windows AD system to other IGI labs. It is using a Windows Server 2016 machine running Active Directory and little else. It has an IP of 192.168.42.69 and two administrative accounts: Administrator and NetworkAdmin (both with password of “Passw0rd”). If running this server locally, you just need to unpack and start the VM. If running on Skytap or SCS-Portal, it is included in the template/instance.

No further lab setup is required.

# 3 Lab 1 – AD Integration with Reconciliation and Provisioning

This lab is built around the requirement for managing accounts and access on Active Directory. It will involve configuration and use of:

- The AD Adapter and the Enterprise Connector module (including reconciliation schedules)
- Account configuration including attribute mapping and enforcement
- Access requests and provisioning

Note that this lab is independent of the other lab. They can be run in any order. This lab needs the Windows Server image as well as the IGI VA and Data Server.

## 3.1 Overview of Scenario

This scenario is summarized in the following table. This is just for your information.

| | |
|---|---|
| ***Summary*** | Configuring AD account attributes linked to person attributes, changing the target attributes then running a recon and forcing a policy enforcement to correct the account attributes (including a rule to drive a trivial person change to trigger the attribute defaults).<br>Configuring an AD adapter with the Enterprise Connectors module and adding some attribute mapping rules. |
| ***Requirement statement*** | *Our main source of concern around identity management is our Active Directory environment.*<br><br>*We need to be able to centrally manage accounts in AD based on users and their roles. This includes provisioning accounts for new users and reconciling accounts to ensure they are within policy.*<br><br>*Many AD account attributes are linked to attributes on the person in HR. If one of the AD admins changes a value in AD, we need to be able to re-apply the correct (person) attribute value.*<br><br>*There are also some complex AD attributes that we need to define with some logic on provisioning.* |
| ***Demonstration requirements*** | Must be able to demonstrate:<br>• User create generates AD account with the correct attribute values<br>• Reconcile of account with changed attributes will re-apply the correct person attributes to the AD account |
| ***Implementation notes*** | This will require configuration of the following:<br><br>1. Target configuration - a new AD target defined in the Enterprise Connectors module<br>2. Account configuration - including attribute management<br>3. Rules - a User (Account) Modify rule to trigger re-eval of the person-account attribute defaults<br>4. Entitlements - default basic AD entitlements associated with specific OUs<br><br>Assume: User mods done in Admin Console, account mods done in AD, admin roles are there (user manager etc.) |
| ***Skills required*** | Students should have a good grasp of the following:<br>1. Target Configuration (Product documentation and AD Adapter Guide)<br>2. Account attribute configuration (B340 trg module)<br>3. Rules (Rules Guide and D2** trg modules)<br>4. Entitlement Management (B320 trg module) |

## 3.2 Lab1 – Part A – Detailed Lab Instructions

The following instructions will walk you through the lab setup and execution. This is the first part of Lab1 and will focus on AD adapter setup and running a reconciliation. The second part of the lab, Part B, will look at provisioning and recon with attribute enforcement.

### 3.2.1 Summary of Lab Flow and Configuration

This lab is focused on the identity management of Active Directory accounts with IGI. It involves setting up the AD adapter (and initial reconciliation), then configuring for and executing two use cases; provisioning with account creation and reconciliation with policy enforcement.

The overall flow for this lab would be:

1. Setup
    a. Check account adoption rules
    b. Create the AD adapter in the Enterprise Connectors module
    c. Check the account configuration for the new AD and setup the account attributes
2. Recon
    a. Setup and run a reconciliation on the new AD
    b. Check the loaded data (accounts and groups)
    c. Check for automatic adoption
3. *Provision (in Lab 1 – Part B)*
    *a. Publish a permission as a default permission and check accounts and access are provisioned*
    *b. Check and run the access request (select a user that doesn't already have an AD account)*
    *c. Review the provisioning flow*
4. *Recon with Attribute Enforcement (in Lab 1 – Part B)*
    *a. Setup a User (Account) Modify rule to trigger a re-evaluation of user-account attribute mapping*
    *b. Change an account on AD and rerun the recon*
    *c. Check that the custom rule has run and that the target changes have been overwritten*

Some demonstration is performed as you walk through the first two steps. Later, in Lab1 – Part B, we configure the components for the third and fourth steps, then run the two use cases.

The areas of configuration that we will explore in this lab are:

- Rules for account adoption
- AD Adapter, Enterprise Connector and EC Reconciliation Schedule

The following sections provide a detailed walk through of the steps to configure and demonstrate the use cases.

### 3.2.2 Checking Account Adoption Rules

Prior to installing the adapter and running the initial reconciliation, we need to check the rules around account adoption. These are the rules that will try to adopt accounts from a recon to existing users in IGI.

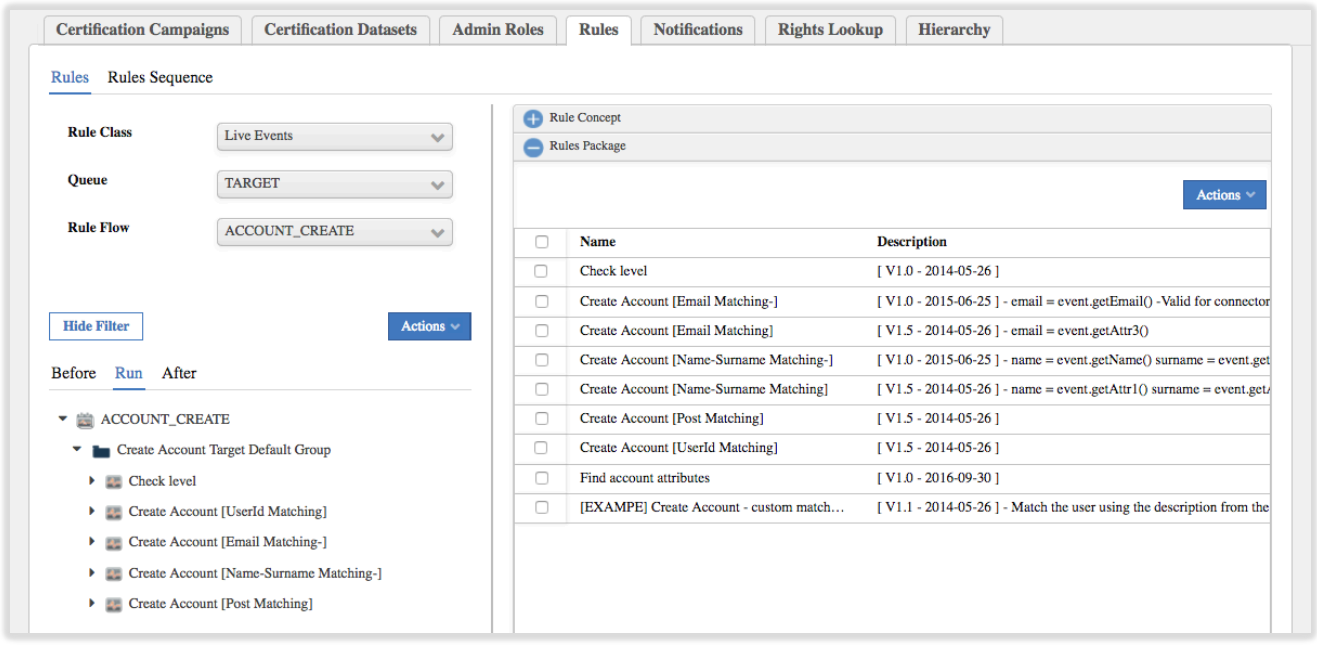
The training system comes with a set of ootb account adoption rules, that will attempt to match (adopt) a new incoming account from a recon with an existing user. These rules run, in sequence, when a new account is returned to IGI from a recon. They will attempt to find a matching user and connect this account to that user.

To check the rules:

☐ Log into the Admin Console (`admin`/`admin`)
☐ Go to **Access Governance Core > Configure > Rules**


The rules are associated with the Target queue, ACCOUNT_CREATE event.

- ☐ On the rules page select **Live Events**, Queue = **Target**, Flow = **ACCOUNT_CREATE**.
- ☐ In the bottom left section, expand the ACCOUNT_CREATE flow to see the rule sequence
- ☐ In the right pane, sales the (**+**) beside Rules Package to see the available rules for this flow.




You can see, based on the rule name, that there is a sequence of rules that will try to match the new account with an existing person by 1) Userid, 2) Email, and then 3) Name-Surname.

Given that the account data in the training system was built from the IGI user data, you can assume that 99% of AD accounts will match by userid. If you want to check the rule code, you select the rule in the right pane and then click Actions > Modify. We are not focusing on rules in this lab, so you don't need to do that.

Now that we know the rules are setup for matching we can configure the adapter and run the initial reconciliation.

## 3.2.3 Install and Configure the Adapter

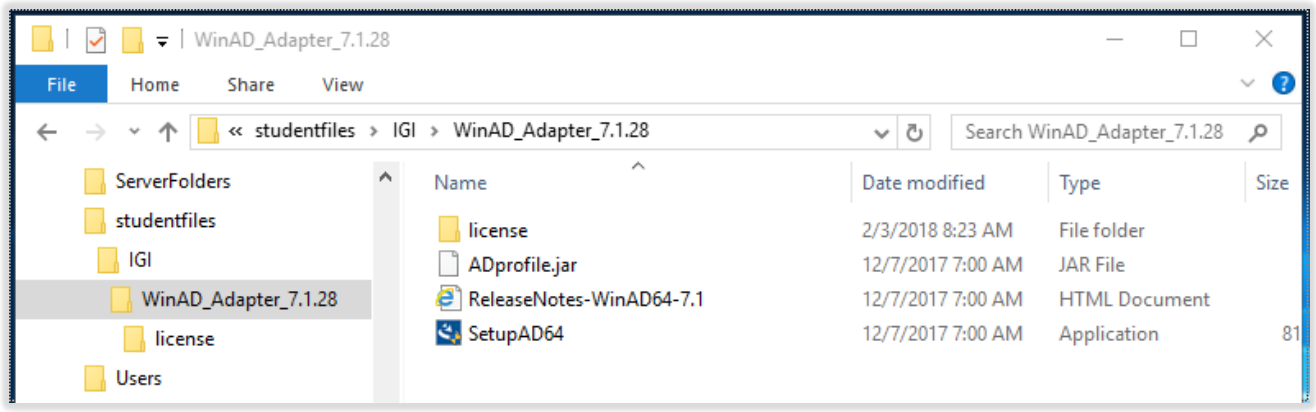
The Windows Active Directory adapter (AD adapter) is one of the older ADK-based adapters from the IBM Security Identity Manager heritage. It does not use Tivoli Directory Integrator – it has an installable agent that is deployed to the AD domain controller (not recommended) or onto another server in the same Windows domain.

Thus, the installation and configuration of the adapter involves:
1. Installing the adapter agent onto the Windows system
2. Installing the adapter profile into the Enterprise Connectors module
3. Creating a new connector for the adapter
4. Configuring and running an initial reconciliation

We will walk through these steps in the next sections.

### 3.2.3.1 Installing the AD Adapter Agent

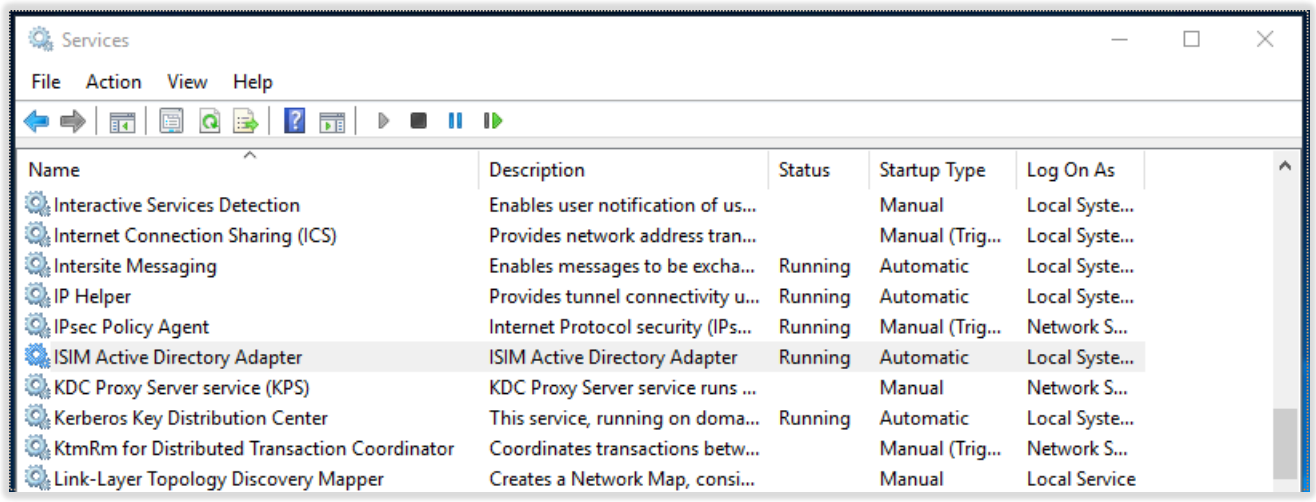
The adapter download includes three files, the agent installer (SetupAD64.exe), the release notes and the profile install (ADprofile.jar).




You would normally download the adapter package, extract the SetupAD64.exe, run it on the Windows server and follow the install prompts. This is very straightforward. This has already been done on the training Windows Server 2016 image.

To check that the adapter is installed and running, perform the following steps (you would normally do this post-install anyway):

☐ Start **Services** and look for the **ISIM Active Directory Adapter**. It should be there and running.




☐ Next, start a **Cmd** window, cd to `c:\"Program Files\IBM\ISIM\Agents\ADAgent"`. There should be files and folders there.

```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\NetworkAdmin>cd "c:\Program Files\IBM\ISIM\Agents\ADAgent"

c:\Program Files\IBM\ISIM\Agents\ADAgent>dir
 Volume in drive C has no label.
 Volume Serial Number is BEAA-E066

 Directory of c:\Program Files\IBM\ISIM\Agents\ADAgent

02/03/2018  08:24 AM    <DIR>          .
02/03/2018  08:24 AM    <DIR>          ..
02/03/2018  08:24 AM    <DIR>          bin
02/03/2018  08:24 AM    <DIR>          data
02/03/2018  08:24 AM    <DIR>          jre
02/03/2018  08:35 AM    <DIR>          log
02/03/2018  08:29 AM    <DIR>          Uninstall_Windows Active Directory Adapter (64 Bit)
               0 File(s)              0 bytes
               7 Dir(s)  26,418,683,904 bytes free
```

The bin (binary) directory is where the agent executables and tools are stored. The data directory is not used with the IGI functions of the agent. The jre folder contains a local java runtime environment (also not used). The log folder contains all of the agent logs which is very useful for problem diagnostics. There is also an uninstaller folder to remove the agent.

☐ Change directory to `bin`
☐ Run `agentcfg -ag ADAgent` (password (configuration key) when prompted is "`Passw0rd`")

```
c:\Program Files\IBM\ISIM\Agents\ADAgent>cd bin

c:\Program Files\IBM\ISIM\Agents\ADAgent\bin>agentcfg -ag ADAgent


Looking for agent 'ADAgent' on node '127.0.0.1'..


Enter configuration key for Agent 'ADAgent':
```

☐ Select **option `A`** to see the current agent settings

```
Configuration Settings
-----------------------------------------
Name                          : ADAGENT
Version                       : 7.1.28 64bit
ADK Version                   : 7.0.3 x64
ERM Version                   : 7.0.3 x64
Adapter Events                : TRUE
License                       : NONE
Asynchronous ADD Requests     : TRUE  (Max.Threads:3)
Asynchronous MOD Requests     : TRUE  (Max.Threads:3)
Asynchronous DEL Requests     : TRUE  (Max.Threads:3)
Asynchronous SEA Requests     : TRUE  (Max.Threads:3)
Available Protocols           : DAML
Configured Protocols          : DAML
Logging Enabled               : TRUE
Logging Directory             : C:\Program Files\IBM\ISIM\Agents\ADAgent\log
Log File Name                 : WinADAgent.log
Max. log files                : 10
Max.log file size (Mbytes)    : 100
Debug Logging Enabled         : TRUE
Detail Logging Enabled        : TRUE
Thread Logging Enabled        : FALSE
```

This shows the agent version (7.1.28 is the latest at time of writing – Feb 2018). Now we need to check the listening port.

- ☐ Select any key to go back to the main menu ("`ADAGENT 7.1.28 64bit Agent Main Configuration Menu`")
- ☐ Select `B. Protocol Configuration`, then `C. Configure Protocol`, then `A. DAML`

```
DAML Protocol Properties
------------------------------------

A.  USERNAME             ***** ;Authorized user name.
B.  PASSWORD             ***** ;Authorized user password.
C.  MAX_CONNECTIONS      100   ;Max Connections.
D.  PORTNUMBER           45580 ;Protocol Server port number.
E.  USE_SSL              FALSE ;Use SSL secure connection
F.  SRV_NODENAME         ----- ;Event Notif. Server name.
G.  SRV_PORTNUMBER       9443  ;Event Notif. Server port number.
H.  HOSTADDR             ANY   ;Listen on address ( or "ANY" )
I.  VALIDATE_CLIENT_CE   FALSE ;Require client certificate.
J.  REQUIRE_CERT_REG     FALSE ;Require registered certificate.
K.  READ_TIMEOUT         0     ;Socket read timeout (seconds)
L.  DISABLE_SSLV3        TRUE  ;Disable SSLv3

X.  Done

Select menu option:
```

The important thing here is the agent listening port, in this case 45580 (this is the default for an agent). We won't explore any of the other options here (see the Adapter Guide for details).

- ☐ To return, type **X** repeatedly until you return to the windows command prompt.
- ☐ Close the command window (we won't need it again)

We have confirmed the adapter agent is installed and working.

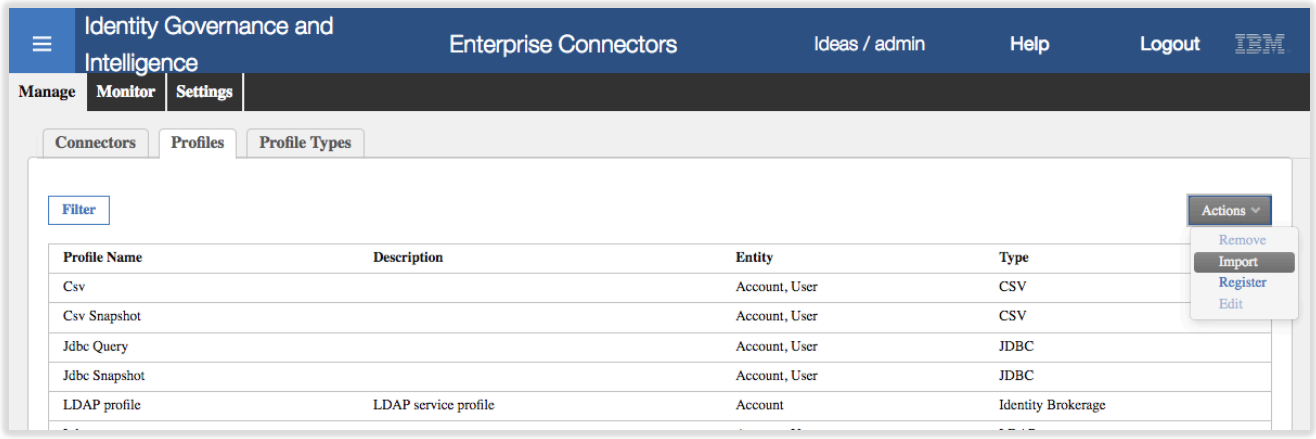
### 3.2.3.2 Installing the Adapter Profile into the Enterprise Connectors Module

Some legacy Enterprise Connectors and Identity Brokerage adapter profiles are pre-loaded into IGI.

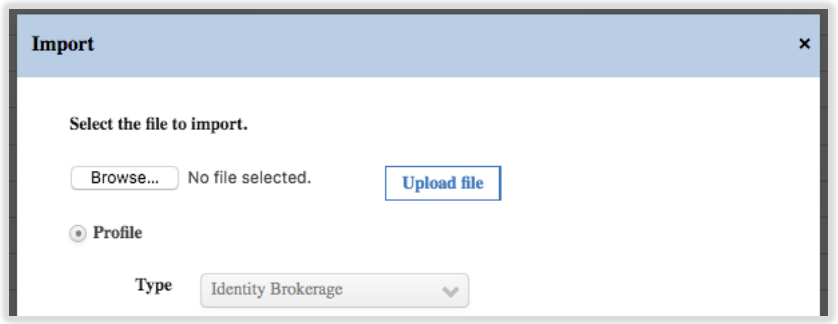
The Windows AD adapter profile is not. It must be loaded before creating a connector for it.

To do this:

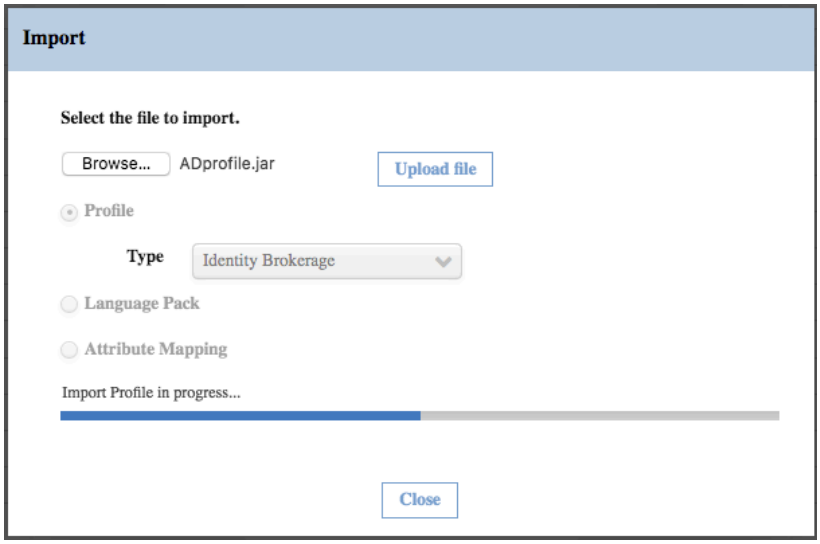
- ☐ Log into the IGI Admin Console (`admin`/`admin`)
- ☐ Go to the **Enterprise Connectors** module
- ☐ In Enterprise Connectors go to **Manage > Profiles**




- ☐ Select **Actions > Import**




- ☐ On the Import dialog, select **Browse..** (may be different label in other browsers) and find the `ADprofile.jar` for the AD adapter
- ☐ Click the **Upload** file button




- ☐ When the import completes, click the **Close** button

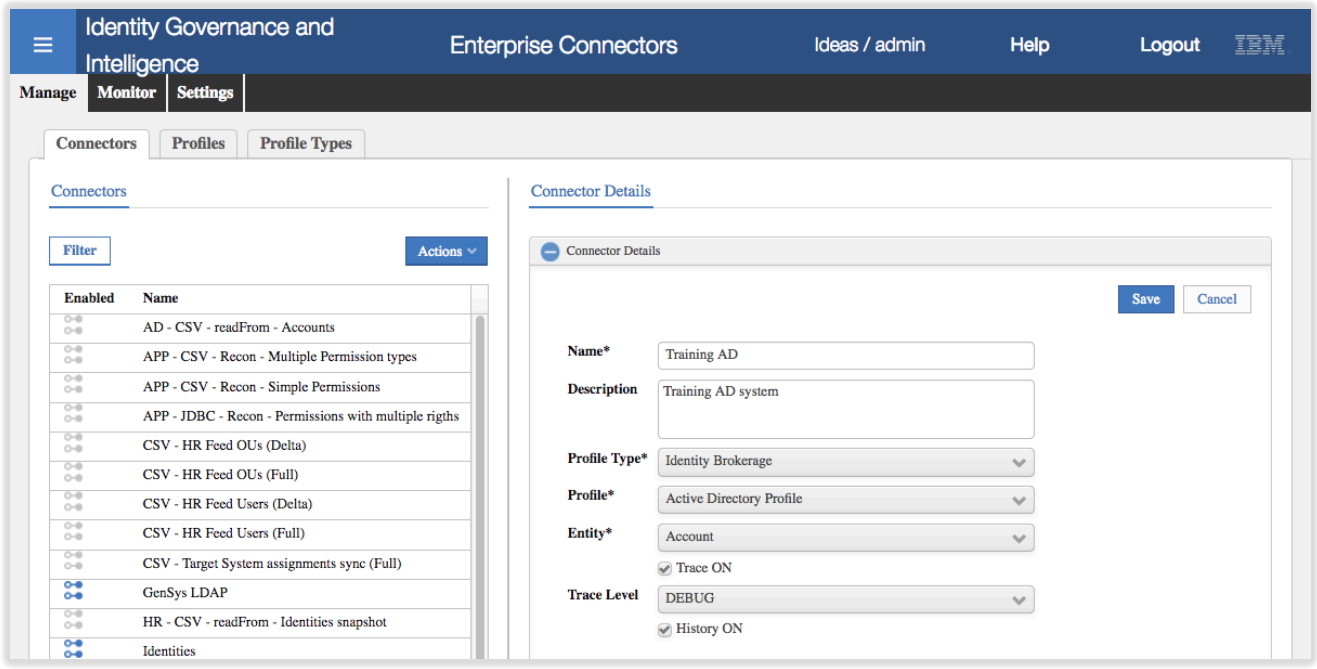
The profile should now show on the Manage > Profiles screen. You can now create a new connector.

### 3.2.3.3 Creating a New Connector for the Adapter

To create the connector:

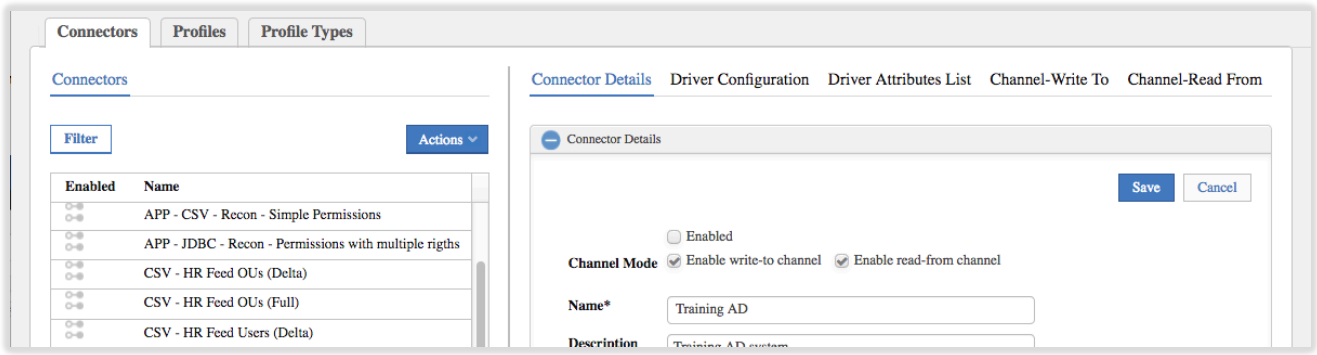
☐ Within Enterprise Connectors go to **Manage > Connectors**
☐ In the left pane, select **Actions > Add**
☐ In the right pane enter the following values:

- Name – Training AD
- Description - <whatever>
- Profile Type – Identity Brokerage
- Profile – Active Directory Profile
- Entity – Account (this will be automatically selected once you select the profile and you can't change)
- Trace ON – Select
- Trace Level – Debug
- History ON - Select




The tracing and history settings are up to you. Given this is the first time the profile and adapter are being used, it makes sense to turn on tracing and set it to DEBUG level. You would turn it down later.

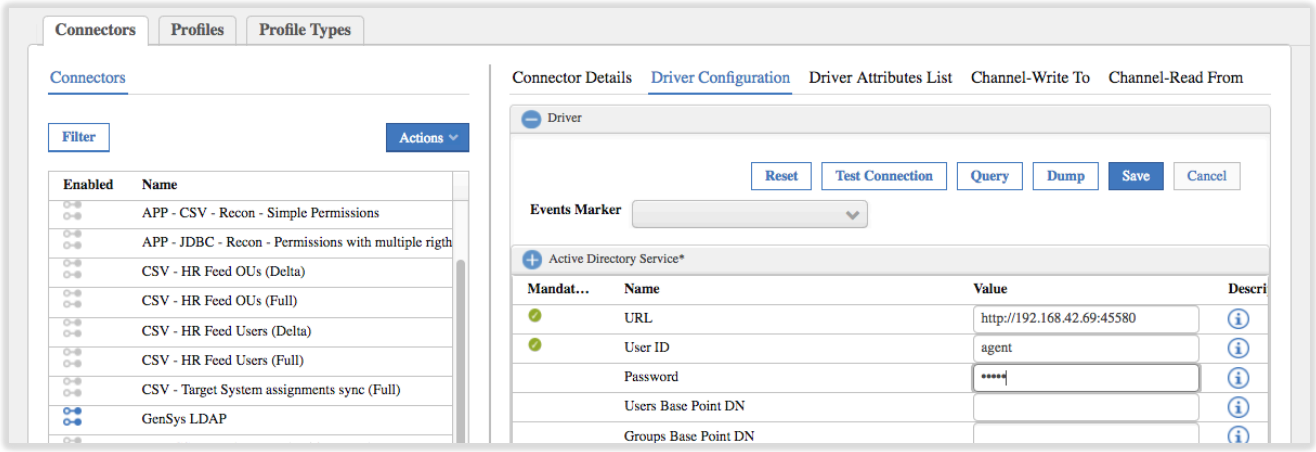
☐ Click **Save**
☐ When the page refreshes, select both the **Enable write-to channel** and **Enable read-from channel** options (do not select Enabled yet)
☐ Click **Save** (this will turn on two more tabs – Channel-Write To and Channel-Read From)



Next, we need to configure the connection parameters, called the Driver Configuration:

☐ Click the **Driver Configuration** tab
☐ Make sure the **Training AD** connector is still selected in the left pane
☐ In the right pane, you need to specify the following parameters in the Active Directory Service section:

- URL – http://192.168.42.69:45580 (this is the IP of the AD server and listening port for the agent)
- User ID – agent
- Password – agent




☐ To test the parameters, click the **Test Connection** button

You should see an Information dialog with "The connection is successful". If you see an error, check your parameters, that the AD server is running and contactable and check that the agent is running. There are commands in the Virtual Appliance command line interface to ping an IP and connect to an IP:port.

☐ Close the **Information** dialog
☐ **Save** the Driver Configuration

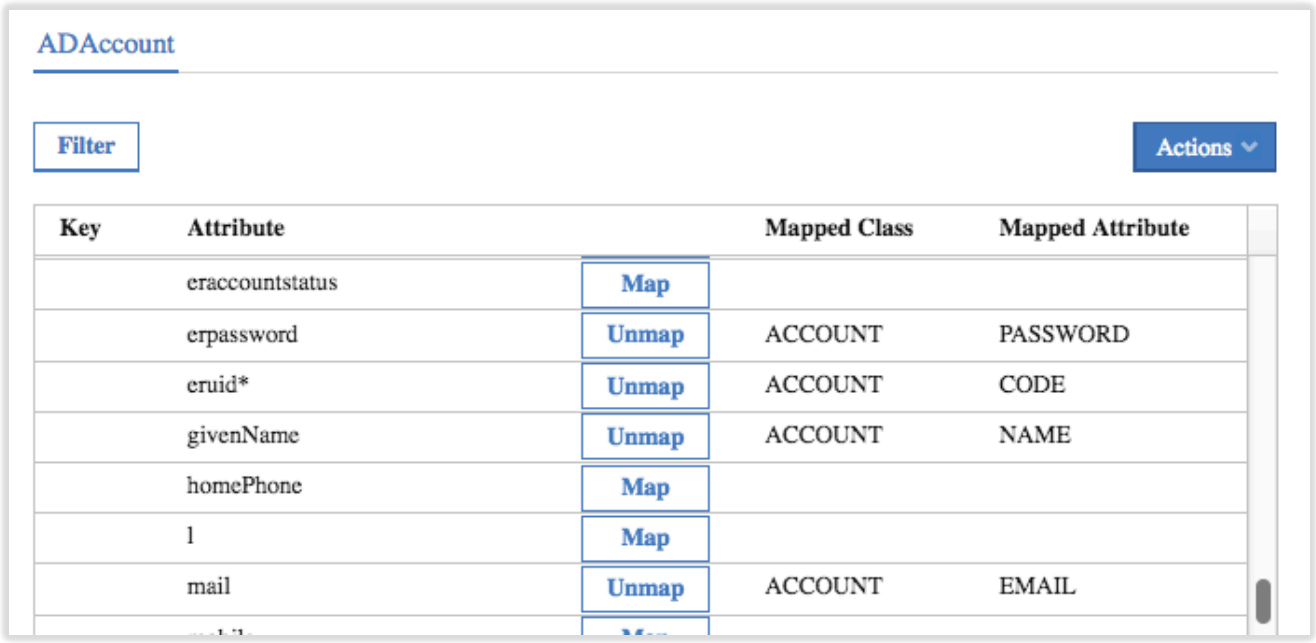
The Driver Attributes List is provided with the adapter profile. For this lab you don't need to change anything.

☐ Go to the **Channel-Write To** tab
☐ Select the mapping icon to see the mapping list
☐ You need to check/set mapping between AD account attributes and IGI generic account attributes. This involves finding the attribute in the list and clicking the Map button to select the mapped attribute. The attributes to be mapped are as follows:

| *Target Account Attribute* | *Mapped (IGI) Attribute* | *Comments* |
|---|---|---|
| erADDisplayName | DISPLAY_NAME | Default mapping |
| erADDistinguishedName | DN | Default mapping |
| erADExpirationDate | EXPIRE | Default mapping |
| erpassword | PASSWORD | Default mapping |
| eruid | CODE | Default mapping |
| mail | EMAIL | Default mapping |
| sn | SURNAME | Default mapping |
| givenName | NAME | First name |

> The list of attributes available for mapping at the account level is very limited due to the IGI account model. All extended attributes are stored in the Broker LDAP (or in a separate table in the IGI DB for legacy connectors) and there is no account-account mapping. We will look at the person-account mapping in the next section.

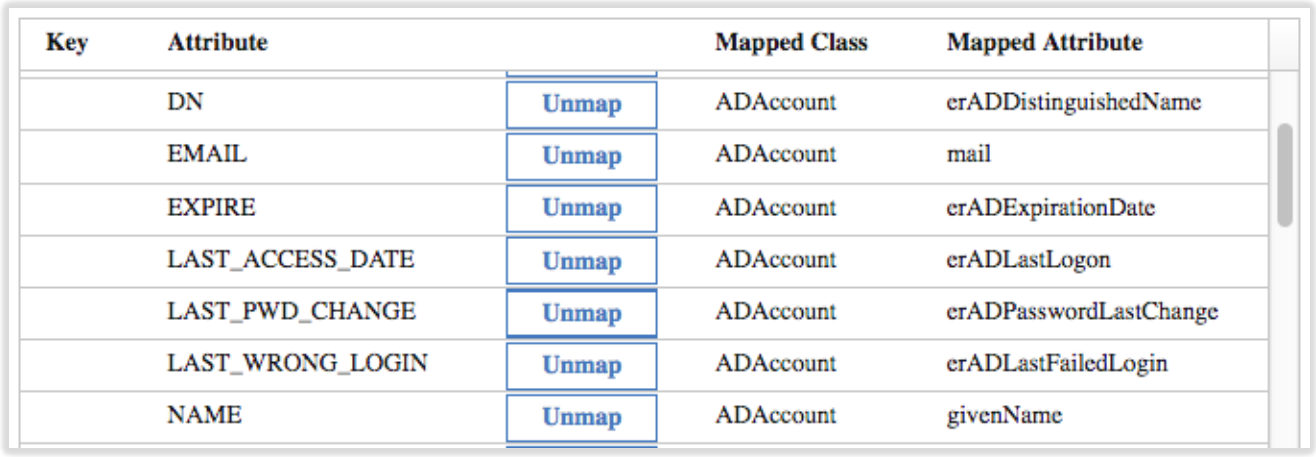
There is no Save function for this – if you've updated the mapping list, it is automatically saved.



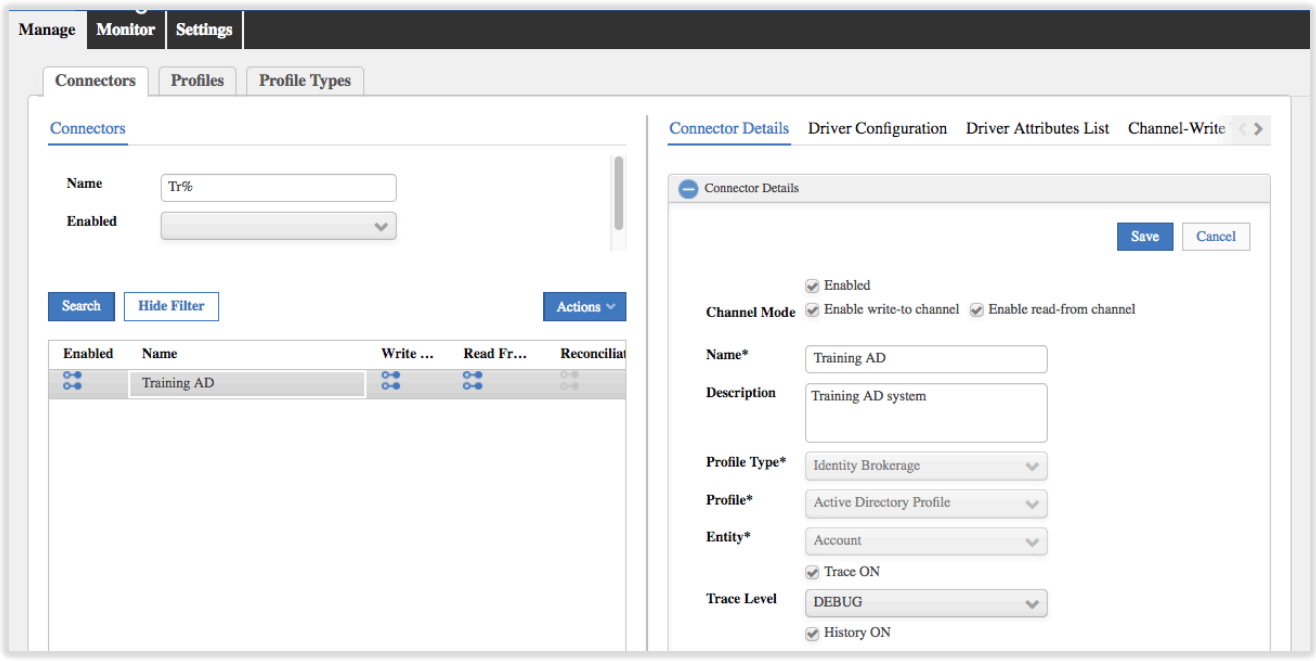

- ☐ Go to the **Channel-Read From** tab
- ☐ Select the mapping icon to see the mapping list
- ☐ You need to check/set mapping between IGI generic account and AD Account attributes. This involves finding the attribute in the list and clicking the Map button to select the mapped attribute. The attributes to be mapped are as follows:

| *IGI Account Attribute* | *Mapped (Target) Attribute* | *Comments* |
|---|---|---|
| CODE* | eruid | Default mapping |
| DISPLAY_NAME | erADDisplayName | Default mapping |
| DN | erADDistinguishedName | Default mapping |
| EMAIL | mail | Default mapping |
| EXPIRE | erADExpirationDate | Default mapping |
| LAST_ACCESS_DATE | erADLastLogin | Default mapping |
| LAST_PWD_CHANGE | erADPasswordLastChange | Default mapping |
| LAST_WRONG_LOGIN | erADLastFailedLogin | Default mapping |
| NAME | givenName | First name |
| PASSWORD | erpassword | Default mapping |
| SURNAME | sn | Default mapping |

As above, there is a limited set to map and most of them are mapped out-of-the-box.



☐ Go back to the **Connector Details** tab and select the Enabled checkbox.
☐ Click **Save**




It shows as enabled (blue icon to the left of the connector name, and it is using both Write To and Read From channels.

We are now ready to configure the adapter for operation and run a recon.

### 3.2.3.4 Configuring and Running an Initial Recon

Reconciliation for Identity Brokerage adapters is a two-step process;

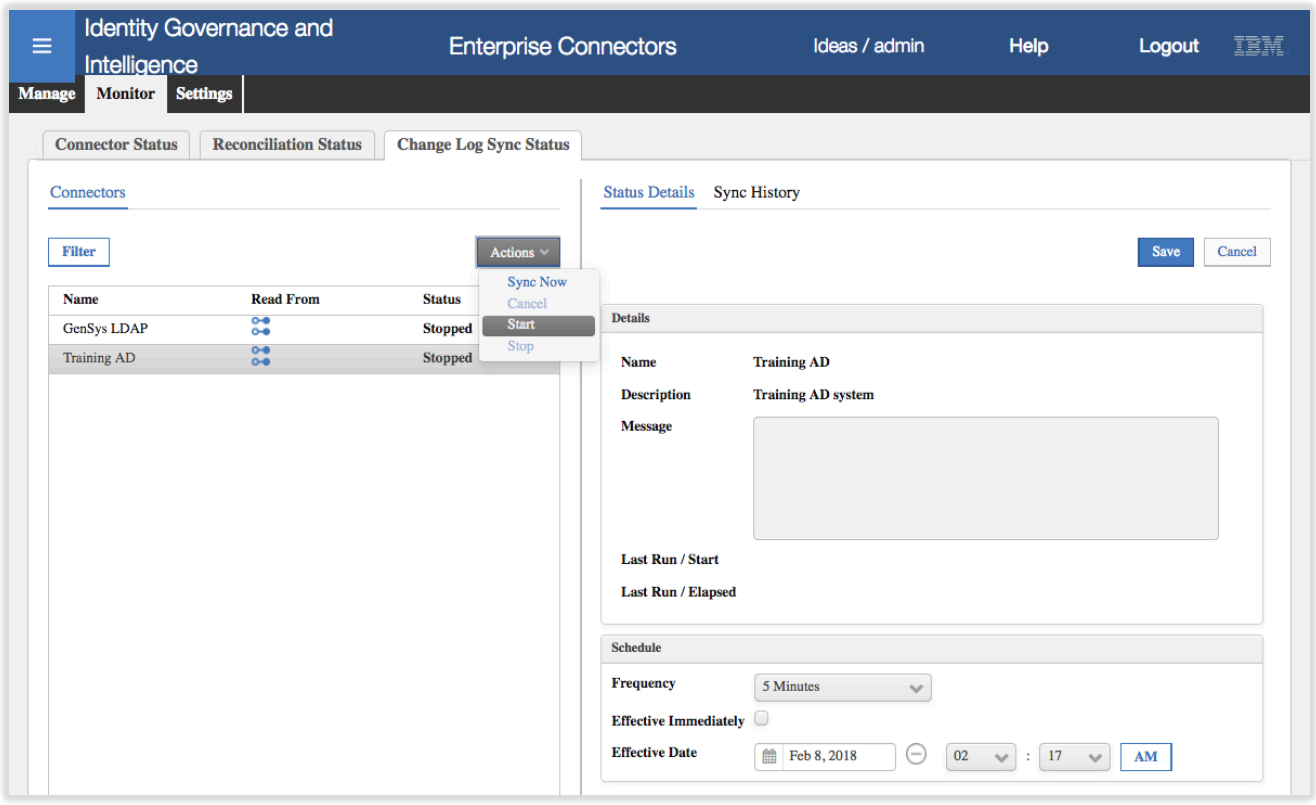
1. The IB will tell the adapter/agent to run a recon, it will run the recon, compare the results with what's in the IB cache (LDAP) and write the changes into a delta table. This is called the "Change Log Sync".
2. The connector will periodically scan the delta table and process any entries there.

In the Enterprise Connectors module, you need to configure two schedules; the one for the change log synch, and the one for the connector Read From processing.

To configure the Change Log Sync schedule

☐ Still in the Enterprise Connectors module, go to **Monitor > Change Log Sync Status**
☐ Select the **Training AD** connector (note that it is in a Stopped Status)
☐ Set the **Schedule Frequency** (in Schedule Details in the right pane) to `5 Minutes` (or similar)
☐ Select the **Effective Immediately** checkbox
☐ Click **Save**

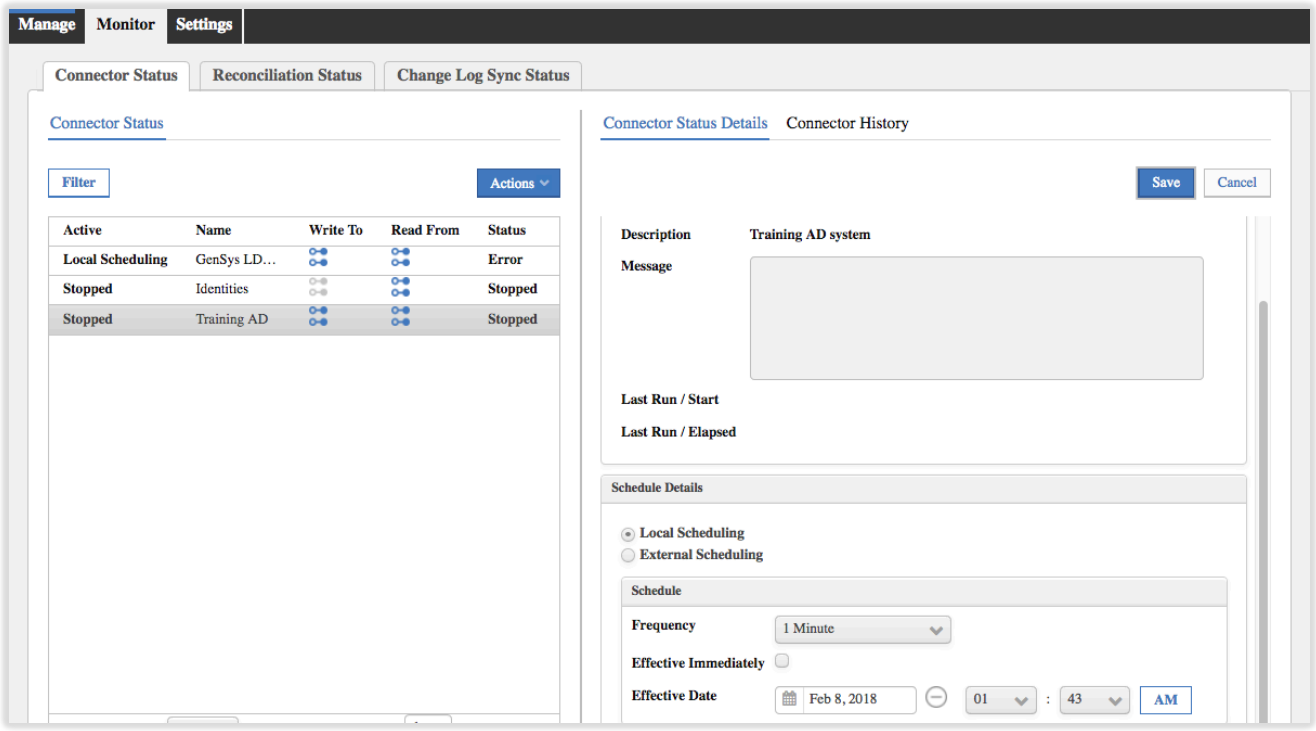
☐ In the left pane select **Actions > Start** to start the sync (the status should change immediately to pending)




We will come back and check on this in a minute.

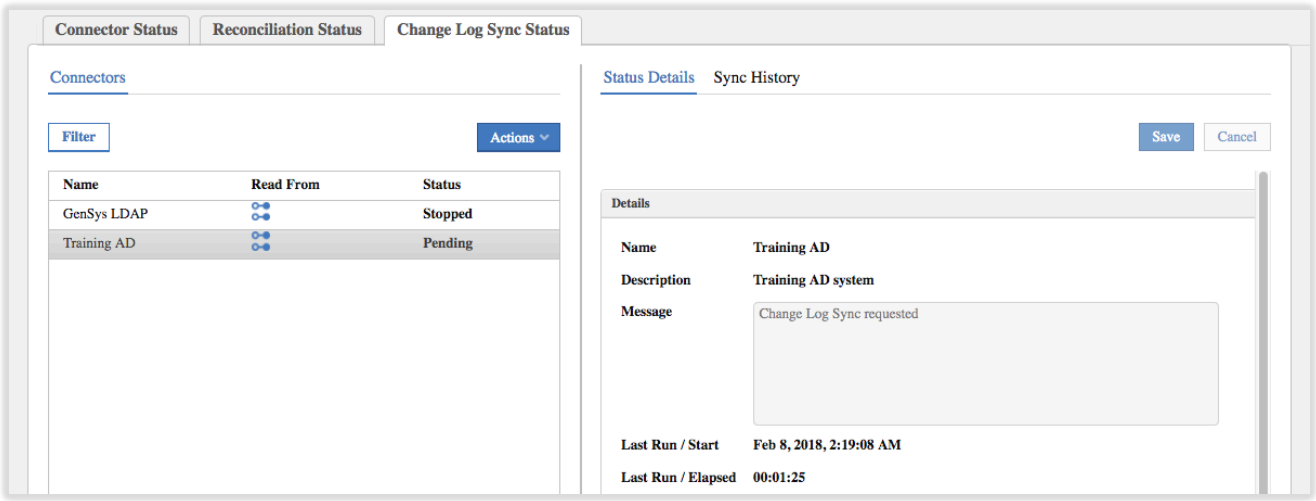
To configure the adapter to run periodically:

- ☐ Go to **Connector Status** and select the **Training AD** connector (note that it is in a Stopped Status)
- ☐ Set the **Schedule Frequency** (in Schedule Details in the right pane) to `1 Minute` (or similar)
- ☐ Click **Save**
- ☐ In the left pane select **Actions > Start** (the status should change immediately to pending)



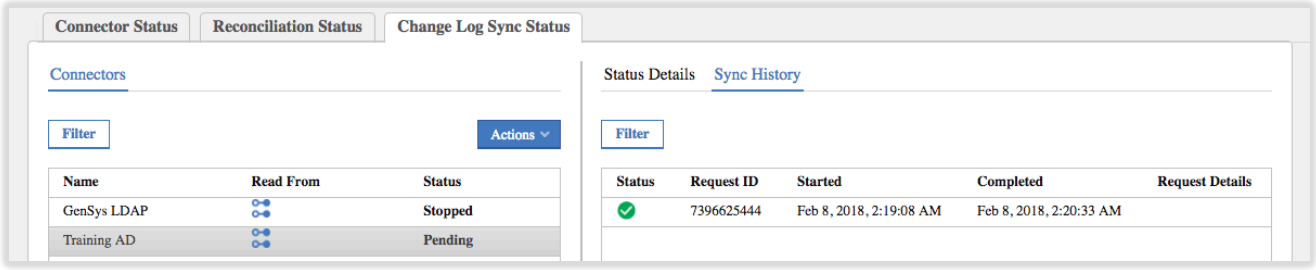
☐ Go back to the **Change Log Sync Status** tab and select the **Training AD** connector

The Status Details view should show a recent execution.




☐ Click on the **Sync History** tab

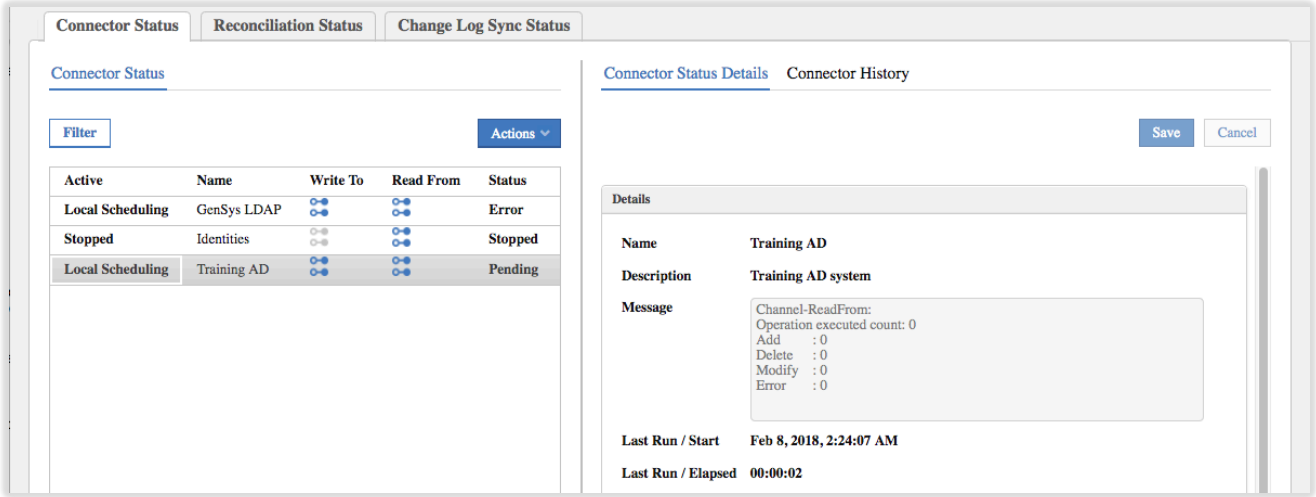
You should see a recent successful execution (if it's showing that it's still running, just click the refresh button).




This means the agent recon has been successfully run and processed by the identity brokerage module. We can now check to see if the connector has picked up the accounts and groups from the recon.

☐ Go to the **Connector Status** tab and select the **Training AD** connector

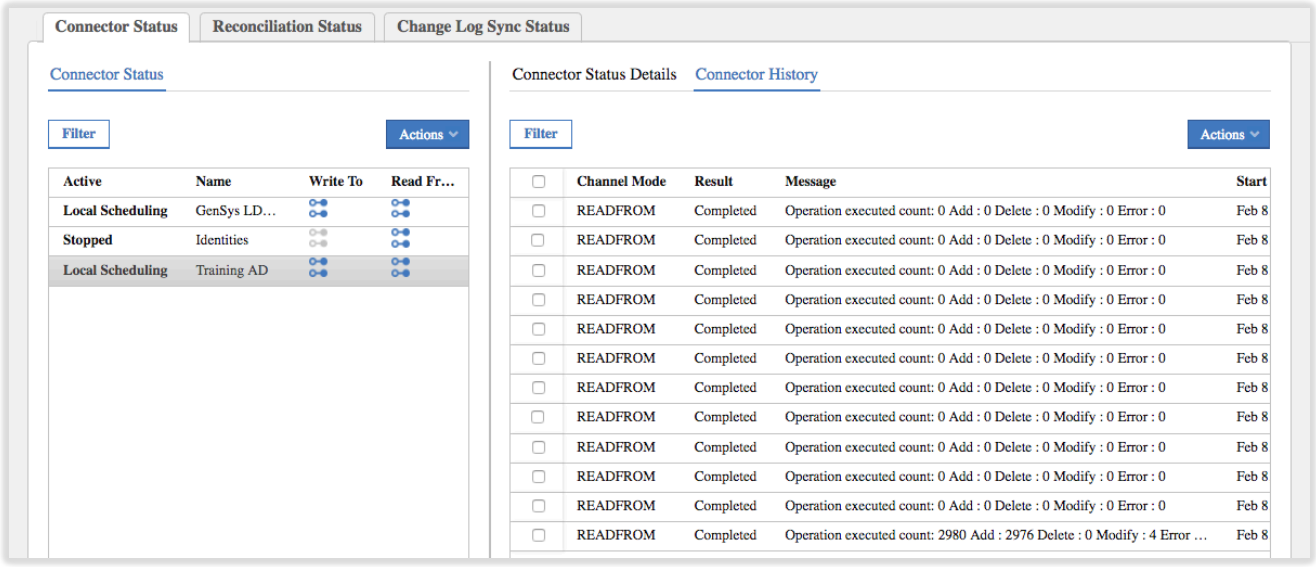
You should see details of the last execution.



It may show information about the load of all accounts and groups from AD in the right pane. However if there have been multiple cycles of the connector, it may show as above.

☐ Go to the **Connector History** view

There should be an execution showing the load of all objects (2980 in total).

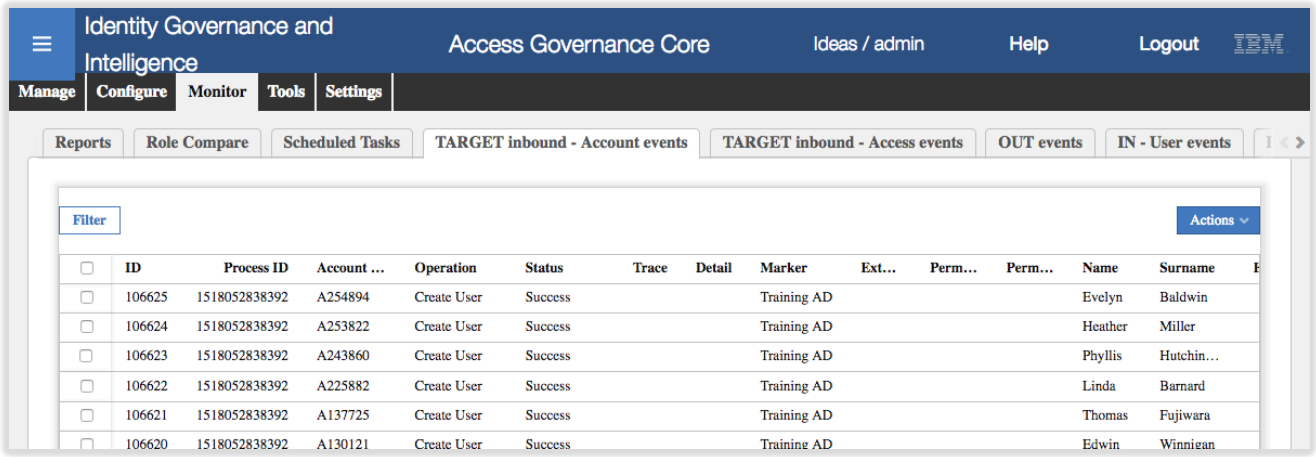



This shows that the connector has consumed all of the accounts and groups from AD.

As a final check we can look at the TARGET queue in Access Governance Core to see the result of this processing.

☐ Go to **Access Governance Core > Monitor > TARGET inbound – Account events**
☐ Click **Filter** and filter on Marker = `"Training AD"`

You should see a lot of entries (2916) consisting of Add Permission and Create User (create account) events. You may need to scroll or page to see the Create User events.




☐ Check also the **TARGET inbound – Access events** to see the group processing
☐ If you see errors, you can select the failed event and click **Actions > Re-execute** (this normally works). If there are any JDBC/DBMS errors on the events, select the event and re-execute (Actions > Re-execute).

We have now setup the adapter and run the initial recon to get accounts and groups into IGI. This completes Lab1 – Part A.

## 3.3 Lab1 – Part B – Detailed Lab Instructions

The following instructions will walk you through the lab setup and execution. This is the second part of Lab1 and will look at provisioning and recon with attribute enforcement for the AD system.

### 3.3.1 Summary of Lab Flow and Configuration

This lab is focused on the identity management of Active Directory accounts with IGI. It involves setting up the AD adapter (and initial reconciliation), then configuring for and executing two use cases; provisioning with account creation and reconciliation with policy enforcement.

The overall flow for this lab would be:

1. *Setup (done in Lab 1 – Part A)*
    a. *Check account adoption rules*
    b. *Create the AD adapter in the Enterprise Connectors module*
    c. *Check the account configuration for the new AD and setup the account attributes*
2. *Recon (done in Lab 1 – Part A)*
    a. *Setup and run a reconciliation on the new AD*
    b. *Check the loaded data (accounts and groups)*
    c. *Check for automatic adoption*
3. Provision
    a. Publish a permission as a default permission and check accounts and access are provisioned
    b. Check and run the access request (select a user that doesn't already have an AD account)
    c. Review the provisioning flow
4. Recon with Attribute Enforcement
    a. Setup a User (Account) Modify rule to trigger a re-evaluation of user-account attribute mapping
    b. Change an account on AD and rerun the recon
    c. Check that the custom rule has run and that the target changes have been overwritten

In this section we configure the components for the third and fourth steps, then run the two use cases.

The areas of configuration that we will explore in this lab are:

- Account Configuration (with Account Attributes), entitlement management and Default access
- Access Request Management workflow
- User Modify workflow (for attribute enforcement)

The following sections provide a detailed walk through of the steps to configure and demonstrate the use cases.

### 3.3.2 Configuring the New AD Account (incl. Attribute Mapping)

Now that we have setup the adapter and loaded accounts and groups from AD, we can configure IGI to support the two remaining use cases; provisioning (with automatic account creation) and reconciliation (with attribute policy enforcement). This and the next few sections will run through these configuration activities.

#### 3.3.2.1 Check AD Account Configuration

When we created the new connector in the Enterprise Connectors module, the process also defined both an application and account configuration to correspond to the connector. We will have a quick look at these.

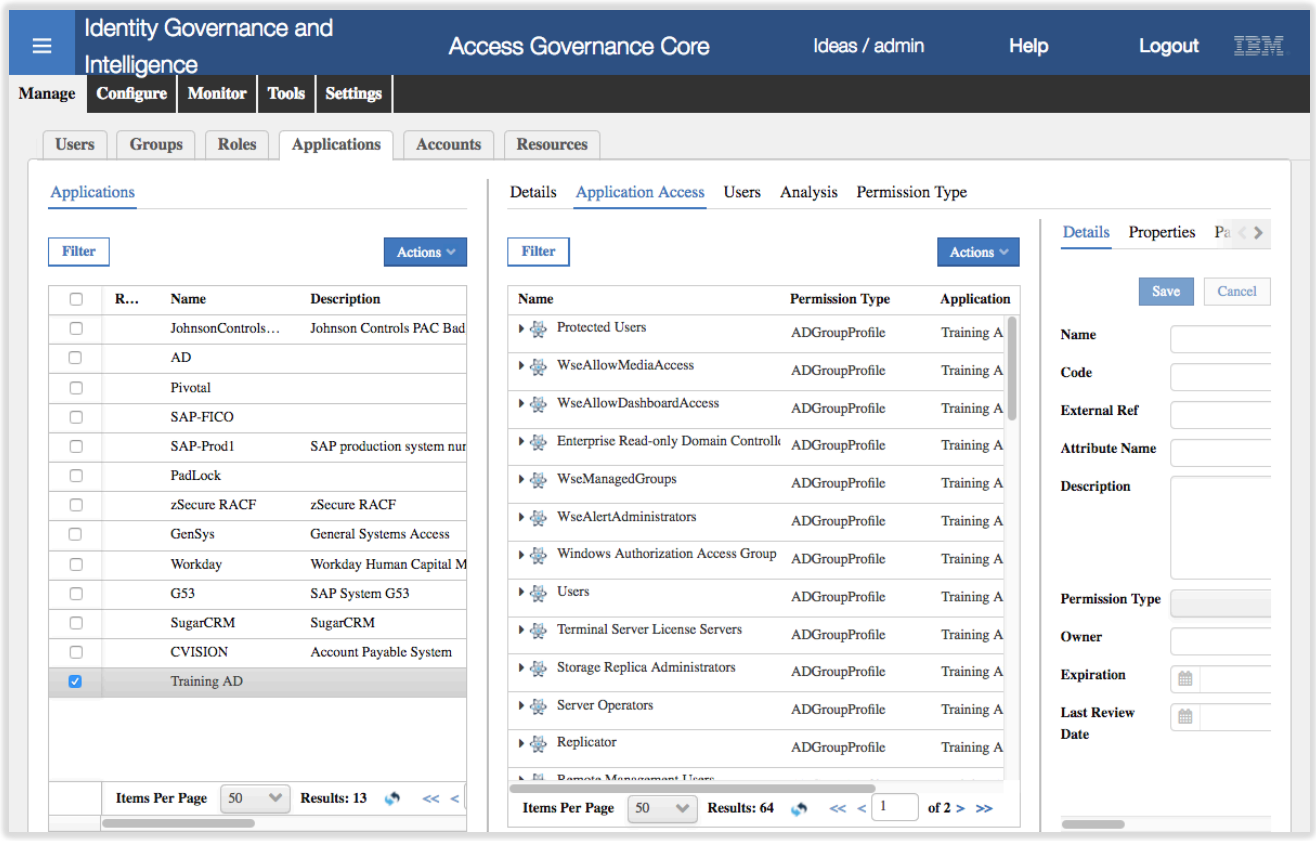
Follow these steps:

☐ Login to the Admin Console (`admin`/`admin`) and go to **Access Governance Core**
☐ Go to **Manage > Applications** and find and select the Training AD application

It has been defined based on the new connector defined. Notice an events marker of Training AD

☐ Select the **Application Access** tab

Here you will see the AD groups loaded from the reconcile. There should be 64 groups.




Notice that the groups are permissions (not external roles – these groups did not have a hierarchy in AD) and are not published by default.

☐ Now go to the **Users** tab

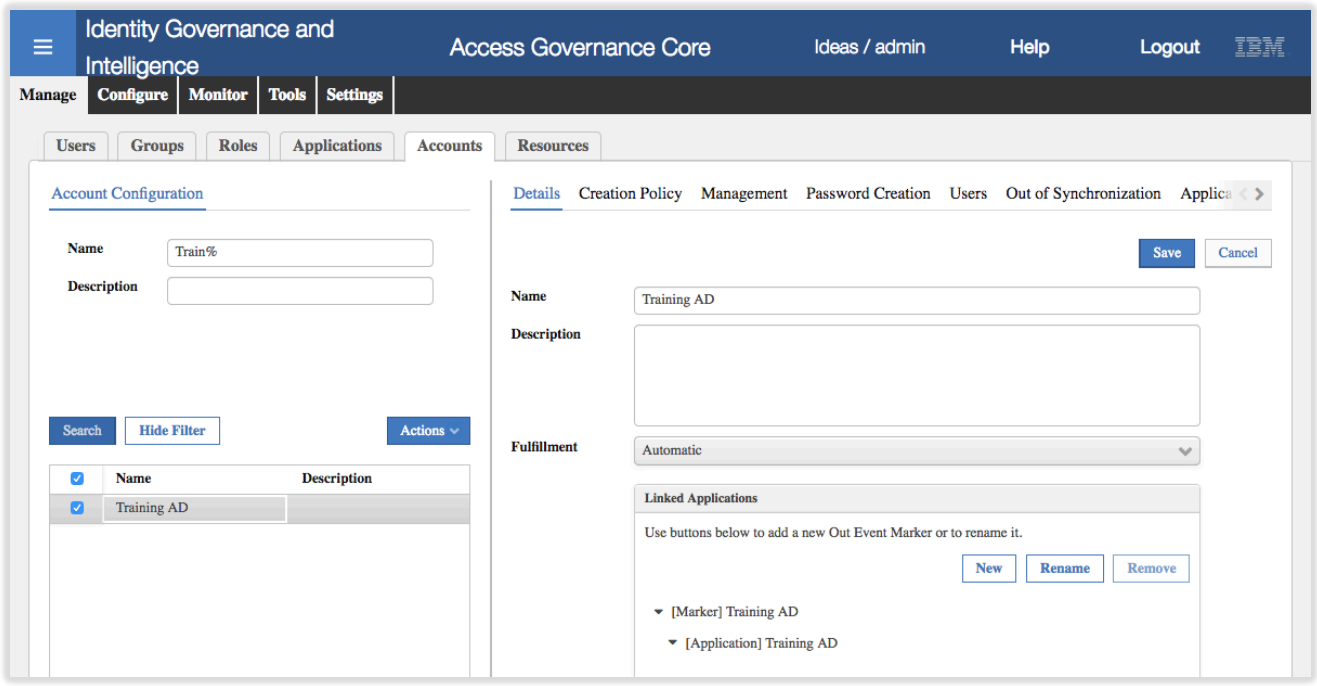
There should be 1367 users listed. This means that of the accounts loaded from the reconcile, 1367 were able to be automatically matched to existing IGI users.

If you found 1362 users in the user list, there was a JDBC/DBMS error on processing some of the accounts or access. It won't really affect the flow of the lab, but you could re-execute those failing events.

Next, we will check the account configuration (account policy):

☐ Go to **AGC > Manage > Accounts**

☐ Find and select the **Training AD** account




The new account configuration is tied to the Training AD application (via the Training AD event marker). The Fulfillment is set to Automatic to any provisioning events sent to the OUT queue will get automatically processed by the adapter.

☐ Go to the **Creation Policy** tab

This is a standard configuration that we don't need to change. Note that the default UserID will be based on the Ideas account ID (which is the Master ID of the person).

☐ Have a look at the **Management** and **Password Creation** tabs

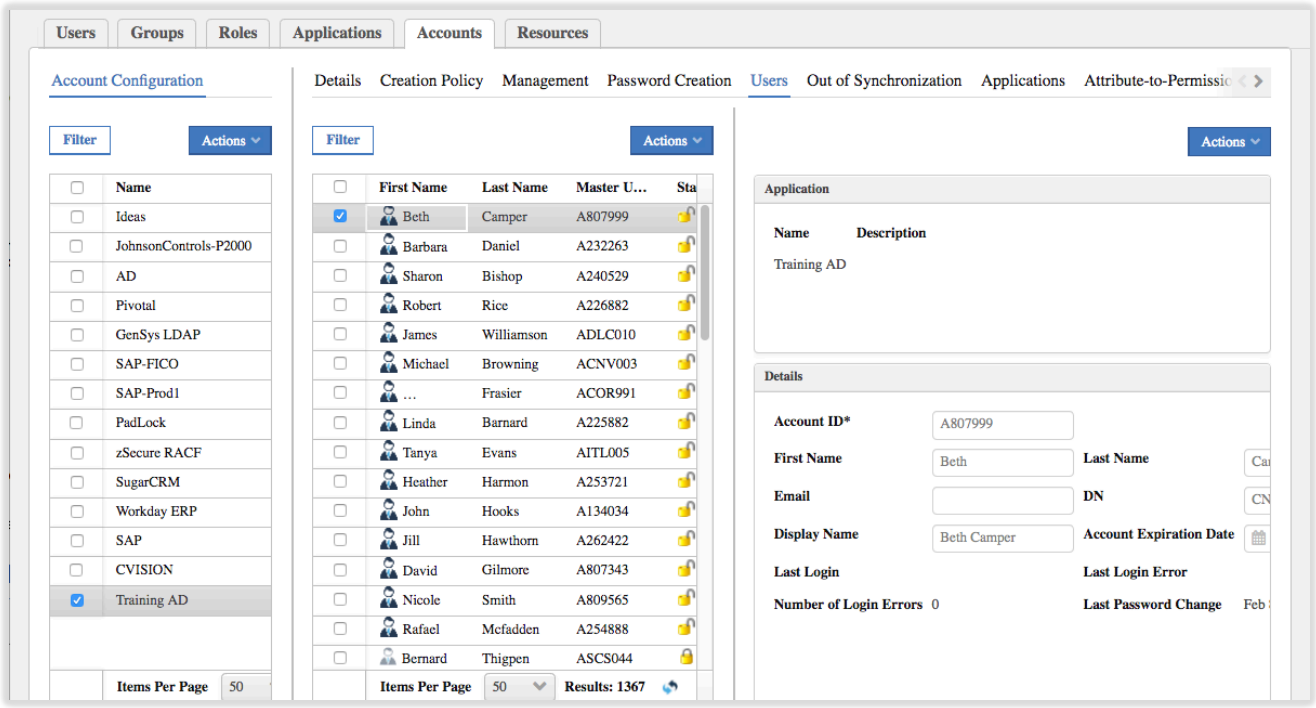
Again, the policy here is fairly standard, and we don't need to modify for this lab's use cases.

☐ Click on the **Users** tab

This lists all of the accounts (even though the tab is called "Users" they are actually accounts).

Notice that there are 1367 accounts, as we saw before.

☐ Select one of the accounts to see the account details (you may need to resize the different panes to see the account attributes)

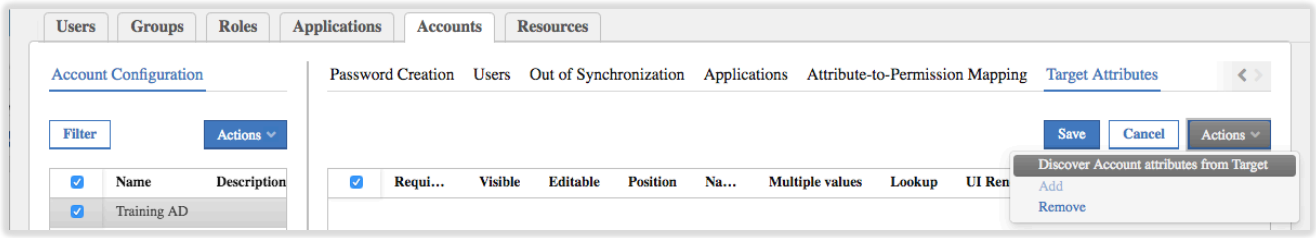



In this case you can see the Account ID, first and last names, email address, AD DN for the account, display name, account expiration date and some login information. We will enhance that list and default mapping in the next section.

### 3.3.2.2 Setup for Account Attribute Mapping

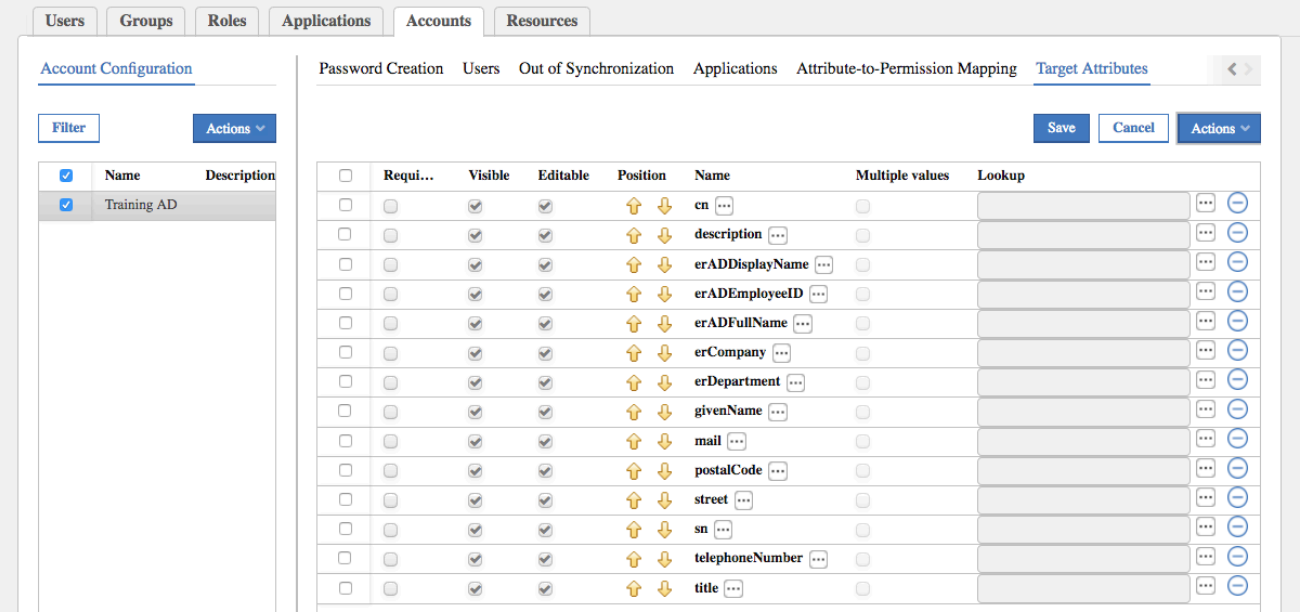
For account creation (and for policy enforcement in the next lab scenario) we need to setup the Target Attributes for this account type. To do this:

☐ Still in **AGC > Manage > Accounts**, go to the **Target Attributes** tab
☐ In the right pane select **Actions > Discover Account attributes from Target**




☐ On the Discover Attributes from Target dialog, select the following attributes: `cn`, `description`, `erADDisplayName`, `erADEmployeeID`, `erADFullName`, `erCompany`, `erDepartment`, `givenName`, `mail`, `postalCode`, `sn`, `street`, `telephoneNumber` and `title`.
☐ Click **Import** when all selected.

The Target Attributes view no shows all of those AD attributes.




These are the attributes that will show in the Admin Console and Service Center and will also be used when IGI automatically creates accounts.

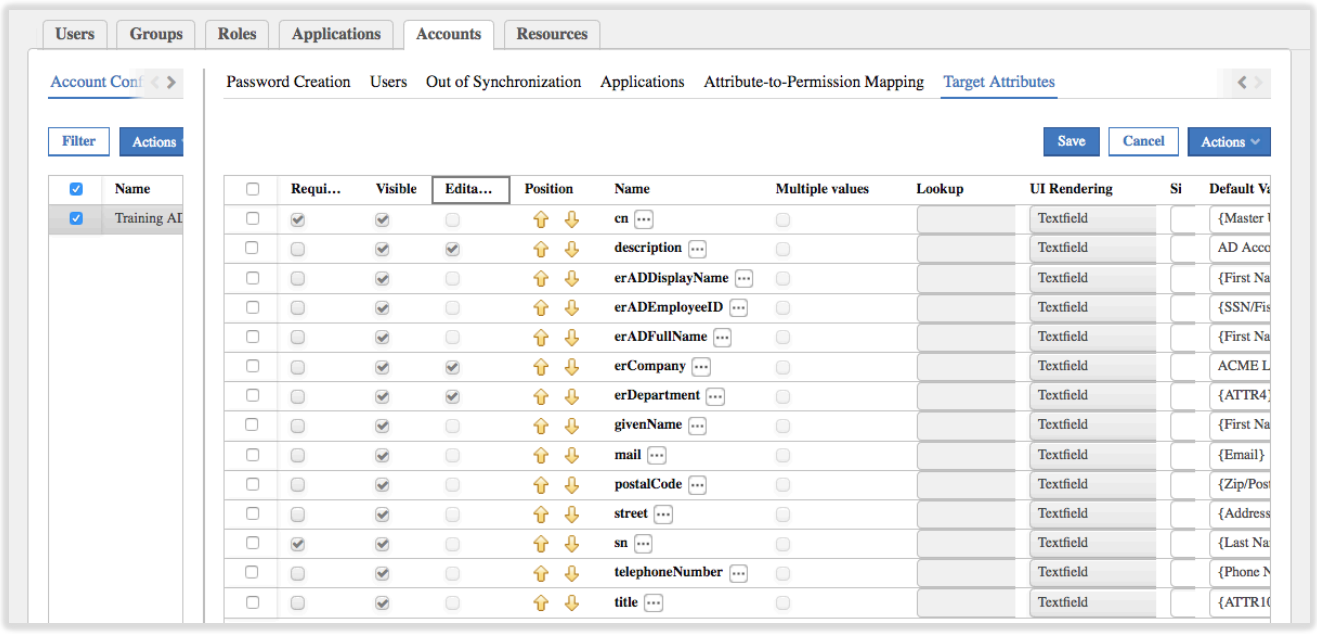
☐ For each attribute set the following:

- **Required** – leave blank for now (except cn and sn)
- **Visible** – leave selected for now (you could have some attributes hidden)
- **Editable** – leave selected for now (will change when we set enforcement)
- **Position** – leave as the alphabetic order for now (you would probably re-arrange in a production deployment into logical groupings like name, address, business details etc.)
- **Name** – for each attribute we want to set the label as per the table below. Click the elipses button ([…]) beside the Name and set the English label. Note that the Labels do not change the Name displayed on this view.
- **Multiple value** – leave unselected
- **Lookup** – not set (we aren't setting any lookup values, but in a production deployment you may do so for fixed values, like titles)
- **UI Rendering** – leave all as textfield
- **Size** – leave as blank (you could change the textfield box length if you wanted)
- **Default Value** – some of these will be based on strings, some on Person attribute values and some a mix. Enter the values as shown in the following table (can use the User.field button)
- **Enforce User value** – as per the following table

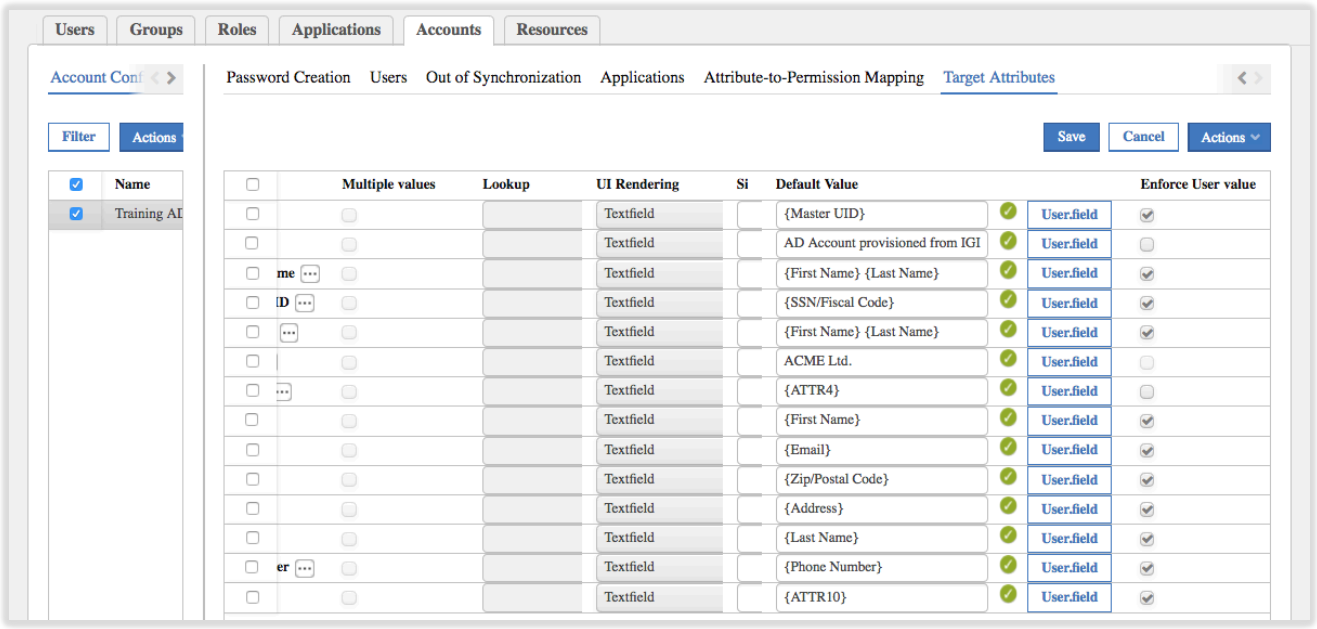
| *Attribute* | *Label* | *Default Value (strings and/or User.field value)* | *Enforced (Y/N)* |
|---|---|---|---|
| cn | Common Name | {Master UID} | Y |
| description | Description | AD Account provisioned from IGI for {First Name} {Last Name} | |
| erADDisplayName | Display Name | {First Name} {Last Name} | Y |
| erADEmployeeID | Employee ID | {SSN/Fiscal Code} | Y |
| erADFullName | Full Name | {First Name} {Last Name} | Y |
| erCompany | Company | ACME Ltd. | |
| erDepartment | Department | {ATTR4} | |
| givenName | First Name | {First Name} | Y |
| mail | Email Addr | {Email} | Y |
| postalCode | Zip/Postcode | {Zip/Postal Code} | Y |
| sn | Last Name | {Last Name} | Y |
| street | Street | {Address} | Y |
| telephoneNumber | Phone Number | {Phone Number} | Y |
| title | Title | {ATTR10} | Y |

☐ **Save** the changes

The result should look like this (with a bit of column resizing)




Scrolling to the right…




We have now set AD account attributes to be defined based on corresponding user account attribute values (or strings/combinations) and also set that most of them will be enforced in the right circumstances (more on this later).

FYI, there are many person attributes we can access to use in defining target attributes. The following table gives a list with examples (for our training system). ATTRnn may be different in other deployments. If you have extended the UserERC schema with your own attributes, you can access them as well. You could also use custom attributes and populate them in mapping rules in the connector (see the pre- and post-mapping rules in the different channel modes in the Enterprise Connector module.

| User Attribute | Contains | Example |
|---|---|---|
| USER_TYPE | User type | Employee |
| OU | Department or Org Unit | Legal |
| PM_CODE | Master UID (Userid) | SChang |
| GIVEN_NAME | First Name | Shirley |
| SURNAME | Last Name | Chang |
| EMAIL | Email | Shirley.Chang@acme.com |
| PHONE_NUMBER | Phone Number | 555 123456 |
| GENDER | Sex | Female |
| BIRTHDAY | Date of Birth | 11/01/1960 |
| ADDRESS | Address (house, number, street) | 1 High Street |
| CITY | City | Hightown |
| *???* | *State (training data uses Nation for State)* | *PA* |
| *NATION* | *Country (training data uses Nation for State)* | *USA* |
| ZIPCODE | Zip/Postal Code | 12345 |
| IDENTIFICATION_NUMBER | CODFISC (we will use for employee id) | AA12345 |
| ATTR1 | Userid of the manager | DFox |
| ATTR2 | Are they flagged as a department manager | N |
| ATTR3 | Highest education level | Upper Secondary |
| ATTR4 | Department | ACME IT |
| ATTR7 | Position | S |
| ATTR9 | Changed | |
| ATTR10 | Title | Solicitor |

We are not using any Attribute Permissions in this lab. Next, we will setup some of the AD groups for accress requests and default assignment.

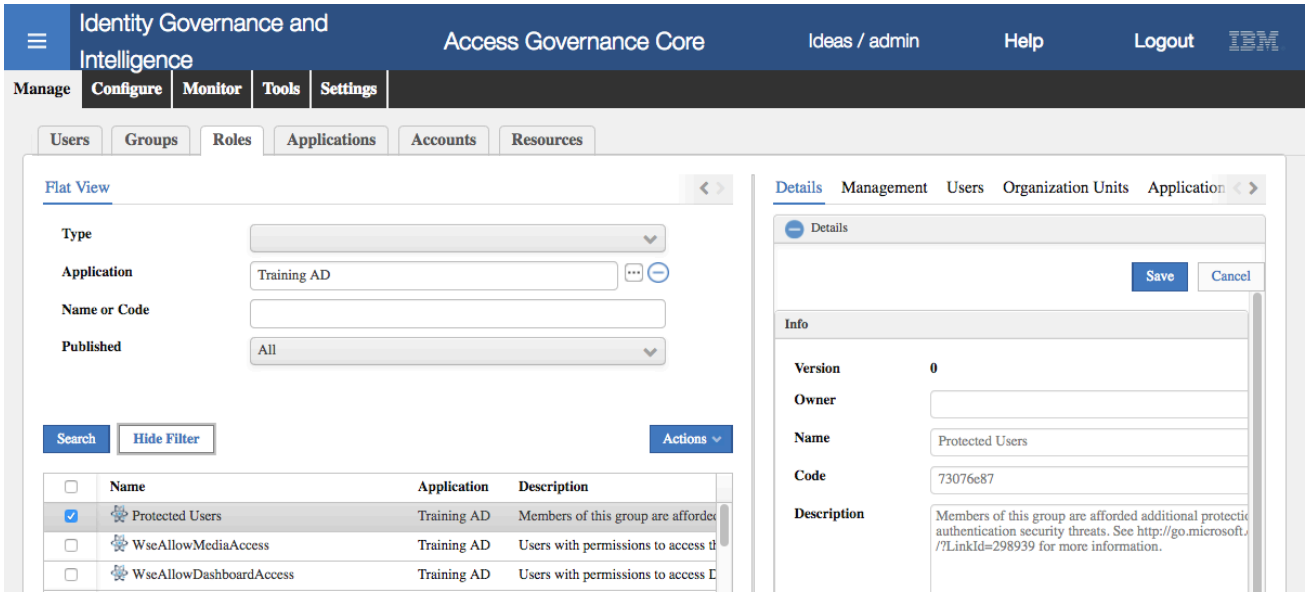
### 3.3.3 Configure Entitlement Management and Default Entitlements

As we saw earlier, we have sixty-four (64) groups in IGI reconciled from the AD system. As noted earlier, they are all sitting there in an unpublished state, which means they are not visible from an access request perspective, nor will that be automatically assigned to any users. We want to fix this for some groups.

To see the group for the new AD system:

☐ In the Admin Console, go to **AGC > Manage > Roles**

☐ Filter to search for **Application** = `Training AD`




You should see the sixty-four groups shown. The names and descriptions have been pulled from AD.

We are going to work on six groups as shown in the following table. You may want to scroll through the list to find each or use the filter function. For each group in the following table:

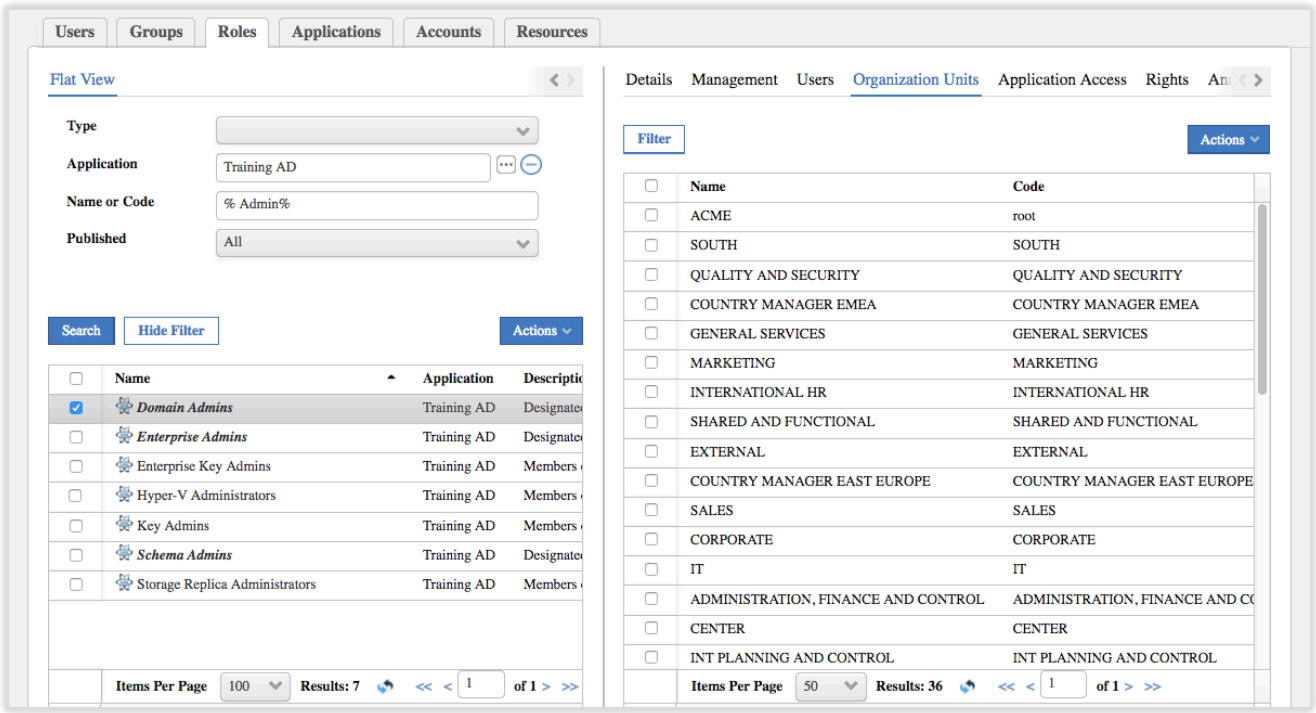
- ☐ Select the group in the left pane
- ☐ Select **Actions > Publish** and click **OK** on the information dialog (note that the group changes to bold+italic font)
- ☐ Go to the **Organization Units** tab in the right pane

> If there was already an account-group mapping for this group, when you publish the group you will see that the organizational unit of the user is already defined in the Organization Units view. This will be the case for NorthRegion and SouthRegion. In this case you do not need to do the next steps.

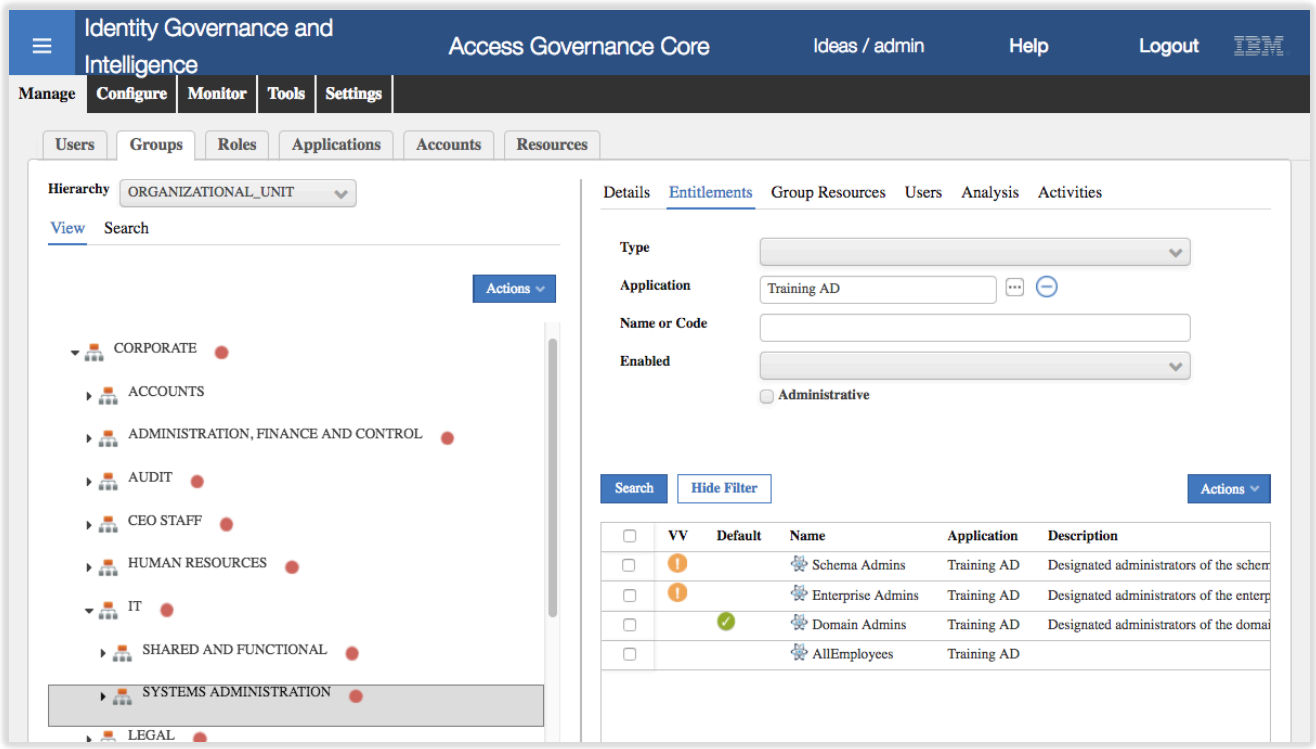
- ☐ Use **Actions > Add** (in the right pane) and on the **Group Selection** dialog, select the org unit listed in the table below and click **OK**
- ☐ When the **Insert Group Entitlements** dialog appears, set the **Default**, **Visibility Violation**, **Enabled** and **Hierarchy** settings as per the table and click OK, then click OK on the Information dialog
- ☐ Repeat for all groups in the table.

| *Group* | *Org Unit* | *Default* | *VV* | *Enabled* | *Hier.* |
|---|---|---|---|---|---|
| NorthRegion | NORTH (in PRODUCT DIVISION) | N | N | Y | Y |
| SouthRegion | SOUTH (in PRODUCT DIVISION) | N | N | Y | Y |
| Domain Admins | SYSTEMS ADMINISTRATION (in CORPORATE / IT) | **Y (& align)** | N | Y | Y |
| Domain Admins | ACME | N | Y | Y | Y |
| Enterprise Admins | ACME | N | Y | Y | Y |
| Schema Admins | ACME | N | Y | Y | Y |

Note that we have two settings for the Domain Admins group. If a user is in the System Administration org unit they will automatically get added to the group (and will not be flagged with a visibility violation). All other users can request the access, but it will be flagged as a visibility violation.

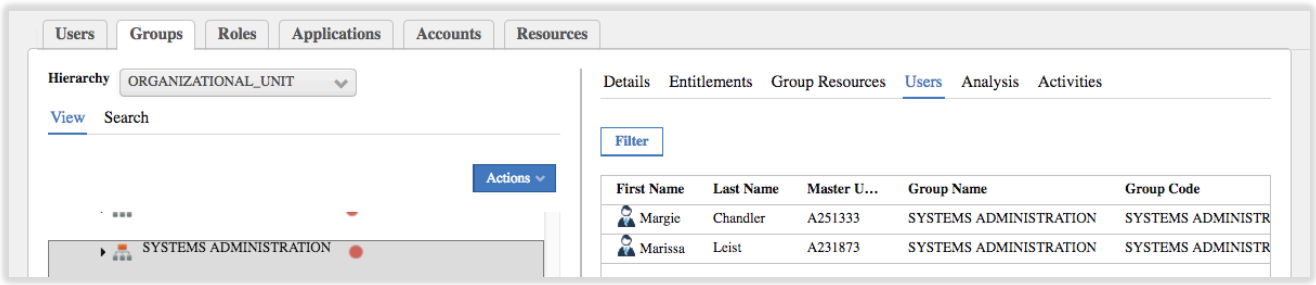



- ☐ Go to **AGC > Manage > Groups**
- ☐ Expand the `ORGANIZATIONAL_UNIT` view to find and select the `SYSTEMS ADMINISTRATION` org unit
- ☐ Go to the **Entitlements** tab and **Filter** on **Application** = `Training AD`

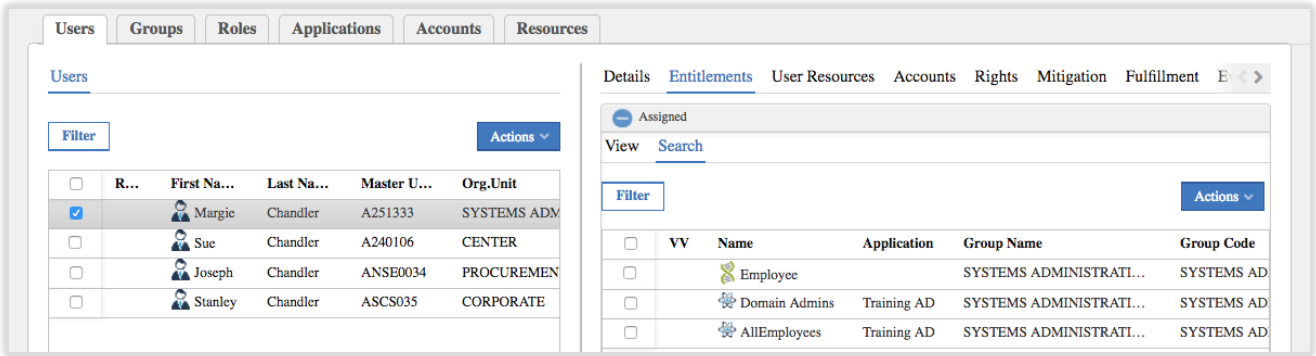



This shows there are four AD groups visible to users in the Systems Administration org unit, and the Domain Admin group is set to default.

☐ Go to the **Users** tab to see the users in `Systems Administration`



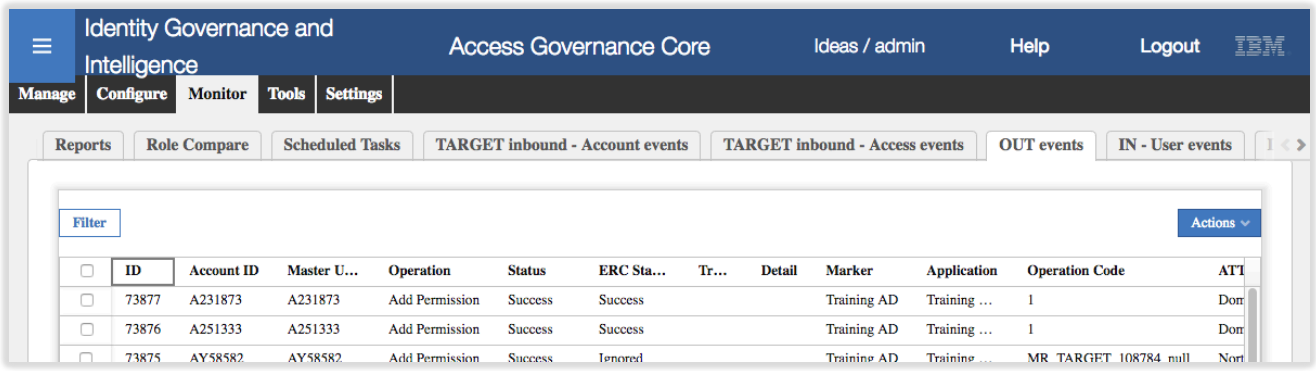

☐ Now go to **AGC > Manage > Users** and find `Margie Chandler`
☐ Confirm that she does now have the `AD Domain Administrators` group




Setting the Domain Administrator group to Default with “Yes, and align users” meant that IGI added that group to any user already in the Systems Administration org unit. This triggered events to the OUT queue to be processed by the adapter.

To check this:

☐ Go to **AGC > Monitor > OUT events**



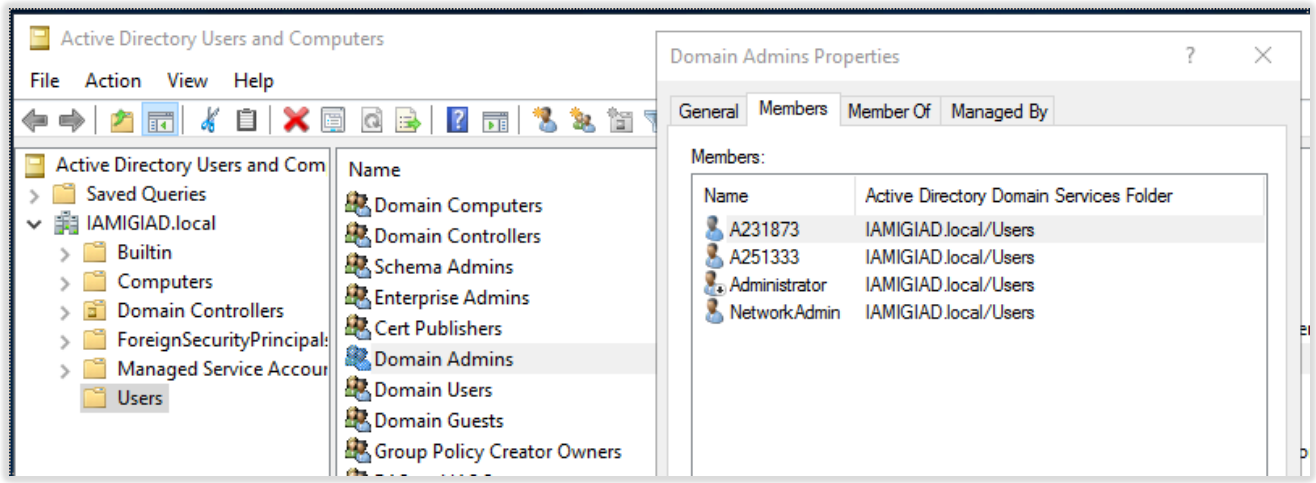

There should be two “Add Permission” events for out two users (Margie A251333, and Marissa A231873).

If the events are sitting in an Unprocessed state for some time, you may need to check for the time drift problem (see the Lab Setup Guide). Often the training image will have problems with time differences between components, leading to events sitting in queues or JDBC commit problems.

If the Status is Success but the ERC Status is in Error and there is a Trace message, there may be a problem with the connector in the Enterprise Connectors module or the adapter itself. You should first check, and perhaps restart, the connector in Enterprise Connectors. With that ok, try to re-execute the events (Monitor view, select the event(s) and Actions > Re-execute). If there is still an issue, you may need to look in the Enterprise Connector, Identity Brokerage or agent logs.

To confirm the provisioning worked as expected:

☐ Log into the **Windows Server UI** (`NetworkAdmin`/`Passw0rd`)
☐ Select **Active Directory Users and Computers** from the taskbar
☐ Expand the tree to go to the **Users** folder
☐ Open the `Domain Administrators` group
☐ Select the **Members** tab




You should see our two users (Margie A251333, and Marissa A231873) listed there.

You can also check the User entries to confirm each are a member of the Domain Admins account.

We have demonstrated part of the provisioning use case. If we were to create a new user in the Systems Administration org unit in IGI, or move an existing user to there, they would automatically be assigned to the Domain Admins group. This is how IGI implements RBAC. In the next section we will setup Access Requests.

## 3.3.4 Configure Access Request Management Workflow

The standard IGI model is that if someone requests an entitlement for a target where they don't already have an account then IGI will generate the event to create the account as well as the event to add the permission to the user. We will leverage this in the provisioning use case.

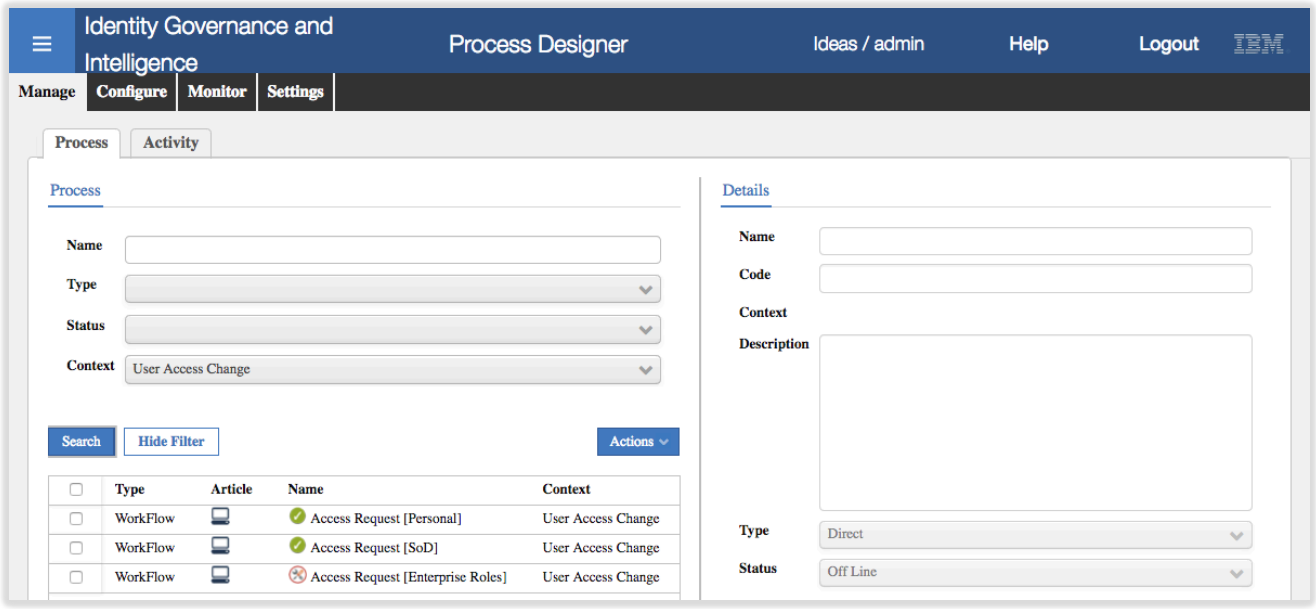
We need to find an existing workflow that meets our needs or build a new one and configure it to support the attribute mapping we defined earlier.

### 3.3.4.1 Exploring Access Request Workflows

We are after a simple access request workflow where a user requests an access and their manager approves. There should be one like that in the training environment.

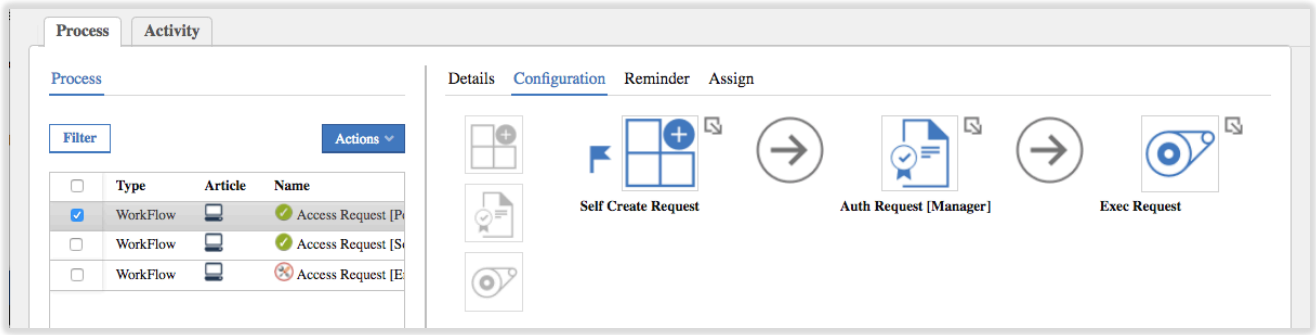
To explore:

- ☐ Log into the Admin Console (`admin`/`admin`) and go to the **Process Designer** module
- ☐ Go to the **Manage** tab
- ☐ Filter on a **Context** of `User Access Change`



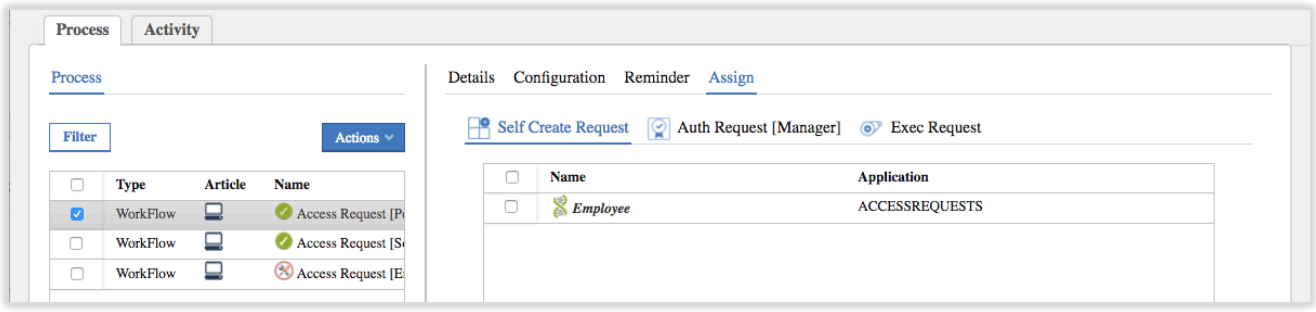

There are three there that may suit.

- ☐ Select the `Access Request [Personal]` workflow process and go to the **Configuration** tab




This certainly looks like what we need; a generate activity ("Self Create Request"), an authorize activity ("Auth Request [Manager]") and an execute request.

☐ To confirm the participants of this workflow, go to the **Assign** tab and check the **Admin Role** assigned to both the "`Self Create Request`" and "`Auth Request [Manager]`" activities.




You should see Employee assigned to the Self Create Request activity, and User Manager assigned to the Auth Request [Manager] activity.

This looks to be an appropriate workflow to use for this scenario.

You can check the other two workflows. You will see that they have similar activities but are assigned to User Manager and Application Manager roles, which is not what we want. We will use the Access Request [Personal] workflow.

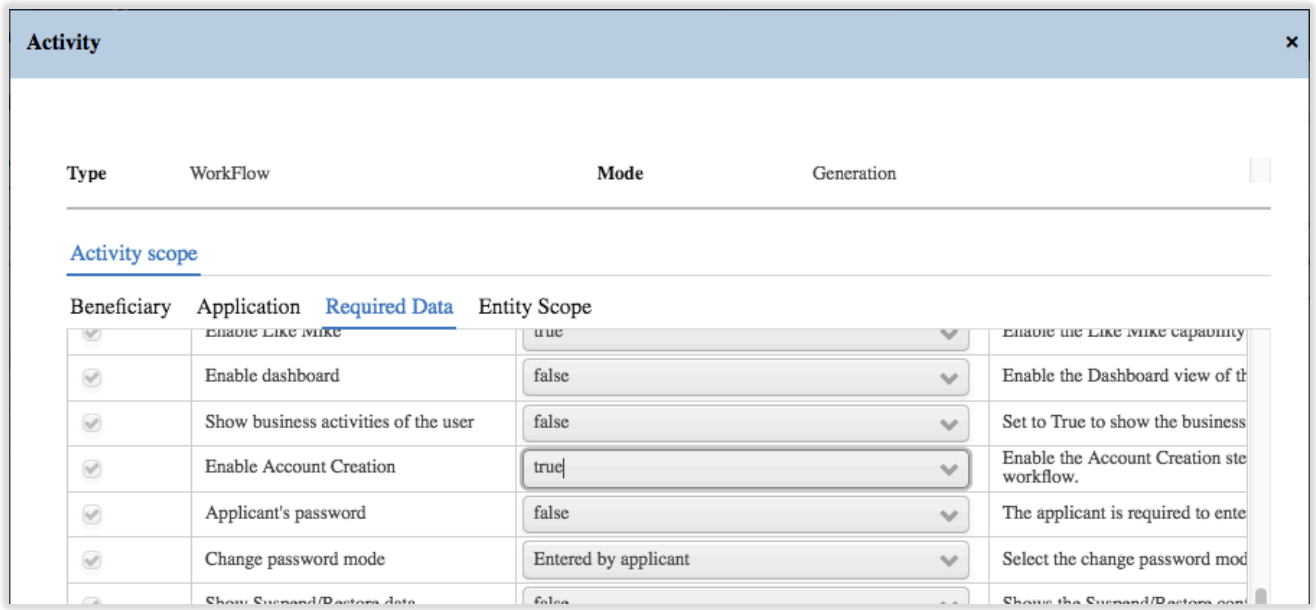
### 3.3.4.2 Configure Workflow

Before changing the workflow, we need to put the workflow into maintenance mode.

☐ Select the Access Request [Personal] process and Actions > Maintenance

> There are some workflow settings that can be changed with the workflow still online, but it's not clear what these are. It's safest to just put the workflow into maintenance mode, make the changes and then bring it back online.

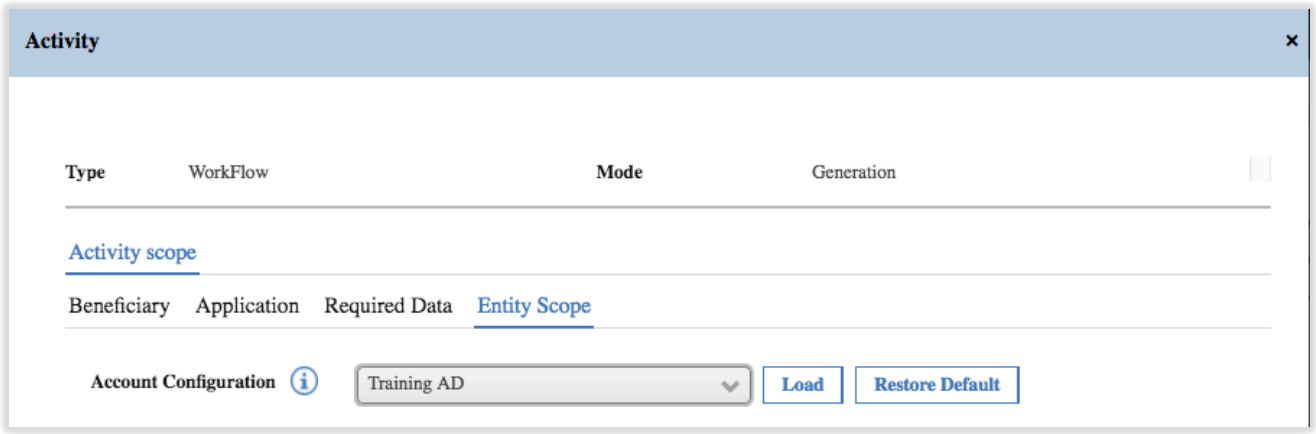
☐ Select the **Configuration** tab and click the `Self Create Request` activity
☐ Check the **Beneficiary** and **Application** tabs. There is nothing to change here.
☐ Click the **Required Data** tab
☐ Scroll down the list of settings to find the **Enable Account Creation** item (you may want to use the horizontal slider to make the lower part of the dialog larger)



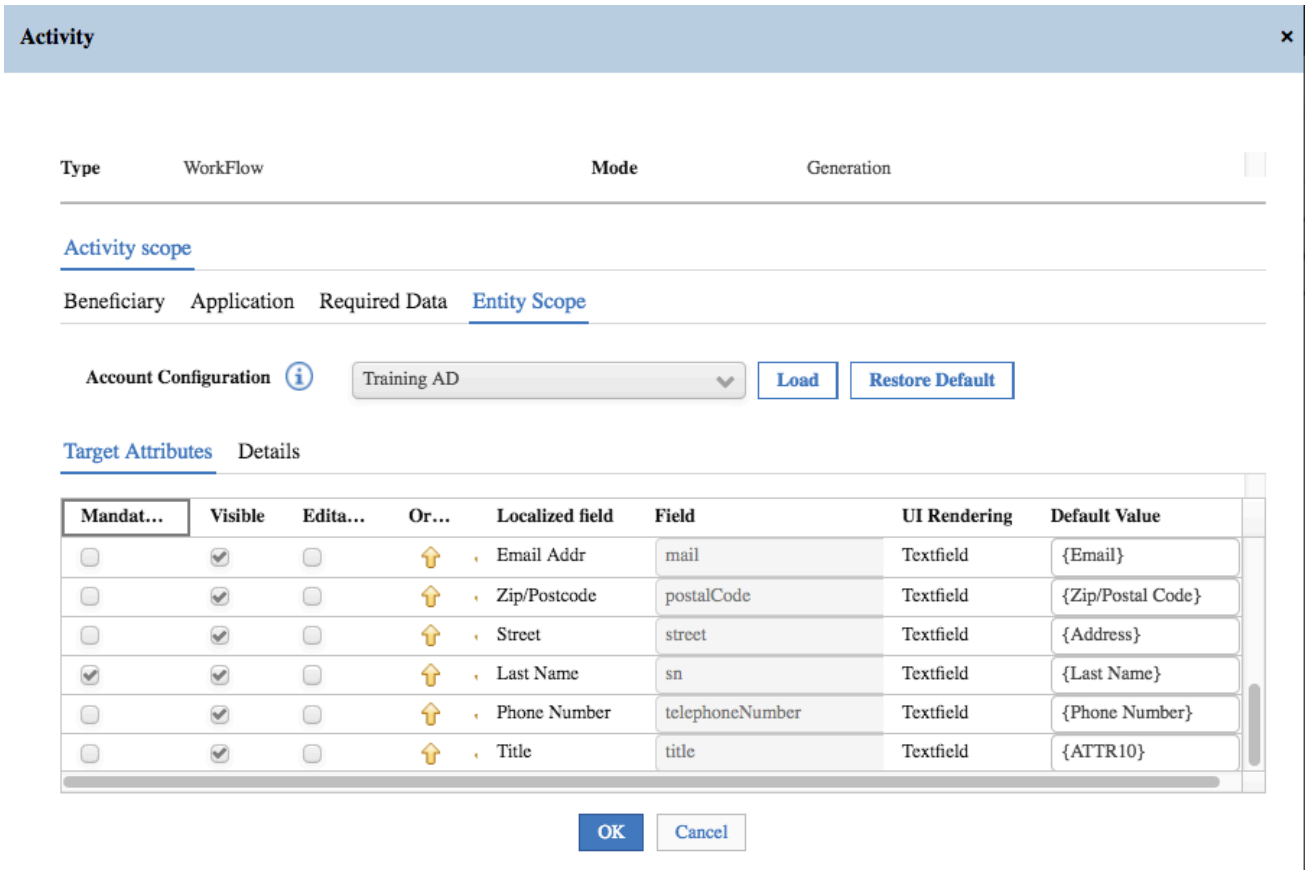
☐ Change **Enable Account Creation** to `true`

This setting will show an extra form in the access request flow specifying account attributes. This will allow us to see the person-account attribute mapping working.

☐ Click the **Entity Scope** tab
☐ In the **Account Configuration** pull-down list, select the `Training AD` application




☐ Click the **Load** button to define the attributes to be presented in the account creation form




We could change the order of attributes displayed and whether they are visible and editable or not. We won't change anything.

☐ Click **OK** to save the changes and return to the workflow configuration screen

☐ Select the `Auth Request [Manager]` activity and have a look at the **Beneficiary**, **Application**, **Entitlement** and **Required Data** tabs

There is nothing to be changed for the beneficiary, applications or entitlements. On the Required Data tab, there is an option to allow the reviewer to edit the account attributes ("Editing of the Account Attributes"). We will leave it set to false.

☐ Repeat the step above in the **Entity Scope** tab to import all of the attributes for `Training AD`
☐ Leave the settings as they are.
☐ Click **OK**

There is nothing to do for the Exec Request activity. As this application (Training AD) is set for Automatic Fulfillment, then the Exec Request node will never be called.

Also, we do not need to change the Reminder or Assignment settings for this workflow.

☐ Select the **workflow** and select **Actions > Online** to bring the workflow back online.

It is now ready to be used to request Training AD access. This completes the configuration required for the provisioning scenario.

## 3.3.5 Configure User Modify Rule (for Attribute Enforcement)

This is the last bit of configuration and is used for the reconciliation (with attribute enforcement) use cases.

As we have seen in earlier sections, we can configure Target Attribute mapping in the Account Configuration so account attributes default to person attribute values, strings or combinations of them. If we create an account in the Service Center or request a permission that also needs a new account, the account attribute values will default to the linked person attribute values (or strings or combinations).

We can also flag those mapped Target Attributes as enforced ("Enforce User value" setting for each Target Attribute). This means that if a user is modified in the Admin Console or Service Center, IGI will re-evaluate any enforced mapping and send a modify event for that account to the target system.

For this user scenario, we want any enforced mapping when we run a reconciliation. Say that we have an attribute set from the user driven by a HR feed (such as title or office location) and we want that user value to be applied to our AD system, even if someone changes the account attribute value on the target system. In an ideal world (and hopefully in some future version) IGI would re-evaluate any attribute mapping policy on an account recon (like ISIM does).

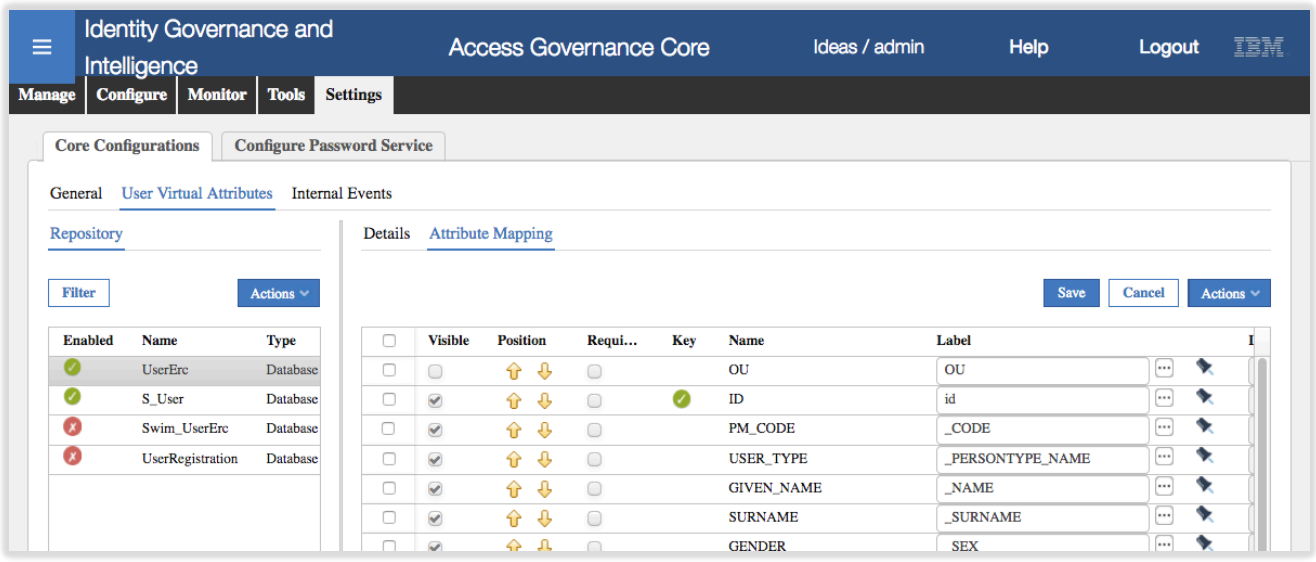
In the current version of IGI we have to trick it into doing the attribute re-evaluation, and we do that by creating a rule that will run when the modified account is reconciled into IGI. This rule will update a person attribute and trigger that re-evaluation. Note that this is currently unreliable – the mechanism sometimes works and sometimes doesn't. There is a fix being developed to resolve it.

The enable this functionality we need to setup a new attribute on the person as our dummy field for update, map it to an unused attribute in AD and setup the rule to modify this attribute.

### 3.3.5.1 Configure an Additional UserERC Attribute

We need an attribute that we can update in a rule to trigger the re-evaluation of attribute policy. We could re-use a currently defined attribute but that carries a risk of breaking something else. You could also add a new attribute to the UserERC schema. However, it is safer and easier to use one of the unused generic ATTRnn attributes. We will use ATTR12.

To do this:
☐ Log into the Admin Console (`admin`/`admin`) and go to **Access Governance Core**
☐ Go to **Settings > Core Configuration > User Virtual Attributes**
☐ Select the `UserErc` database and click the **Attribute Mapping** tab in the right pane



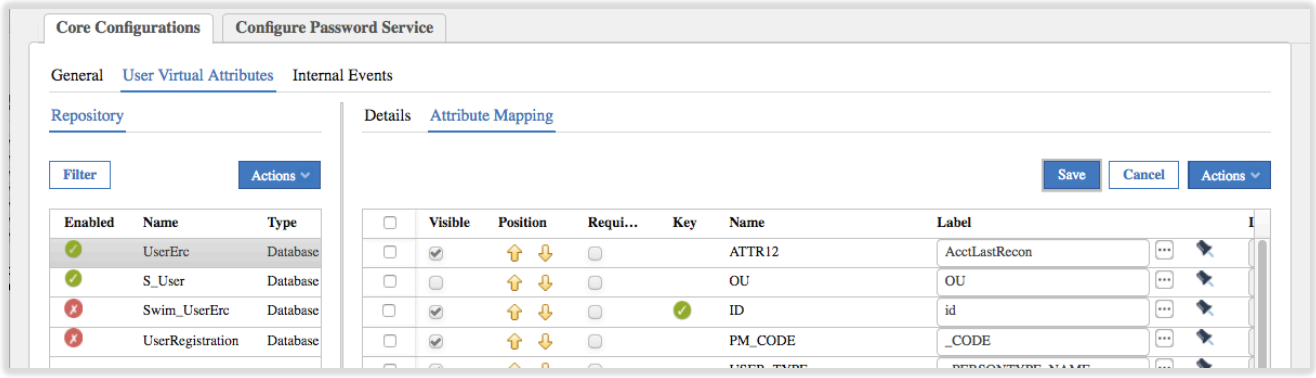

This shows the user attributes currently defined, along with their visibility, order, labels, lookup values, default values and UI rendering. Most of these settings are for user virtual views and workflows, not for the core User ERC view.

ATTR12 is not currently defined so we will define it:

☐ In the right pane select **Actions > Add**
☐ On the **Add Attribute** dialog, find and select `ATTR12` and click **OK**

The attribute is now in the list. You may want to re-arrange the order of the attribute, but it's not necessary for the lab.

☐ Select the new attribute and enter a **Label**. This will be used in the rule and appear on the UI. I would suggest "`AcctLastRecon`" (as we will set a date in the rule).
☐ Click **Save** and click **OK** on the Information dialog box



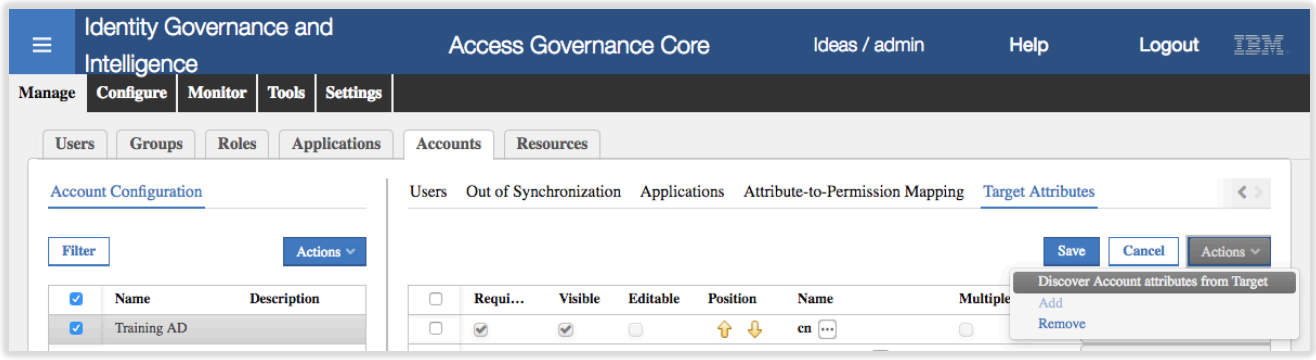

With the new attribute defined, we also need to map it to an unused adapter attribute.

### 3.3.5.2 Map New Person Attribute to Account Attribute

For this workaround to function, we need to map our new IGI Person attribute to an unused AD account attribute and flag it for enforcement.

As you did earlier:

☐ Go into **AGC > Manage > Accounts**
☐ Select the `Training AD` account
☐ Go to the **Target Attribute** tab
☐ Select **Actions > Discover Account attributes from Target**




☐ On the **Discover Attributes from Target** dialog, find and select `erADEExtension1`, then click Import

I am assuming for the lab that this attribute isn't used. For a production deployment you would need to evaluate what free/unused attributes you have in AD.

☐ With the new attribute showing in the **Target Attributes** list, check/set the following
- **Visible** – you may want to hide it from any Service Center data entry workflows
- **Editable** – `disable`
- **Label** – you may want to use the elipses button ([…]) to set a label that makes sense if you leave it visible (like "`Account Last Recon date/time`")
- **Default value** – "`{ATTR12}`"
- **Enforce User value** – `enabled`

☐ **Save** the Target Attributes

We can now create a rule to use this attribute.

### 3.3.5.3 Create the Account Modify Rule

We need to create a new rule that will run when a modified account is reconciled into IGI. Note the code is supplied – you don't need to know Java to do this part of the lab.

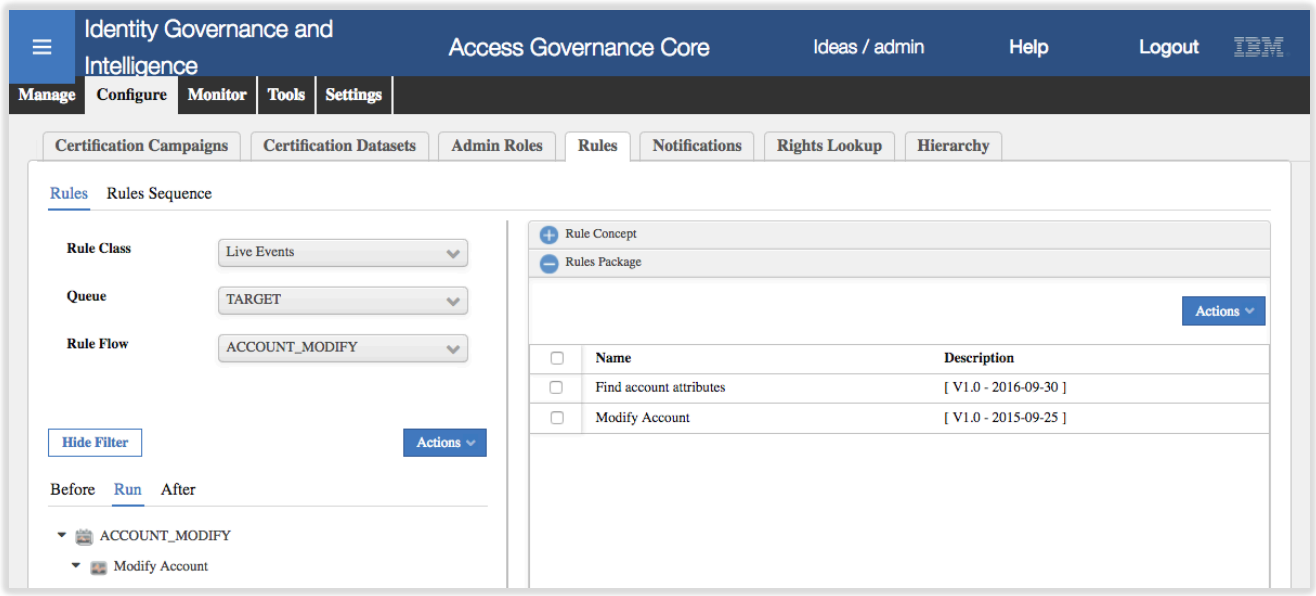
To setup the rule

☐ Go to **AGC > Configure > Rules**

☐ In the left pane select **Rule Class** = `Live Events`, **Queue** = `TARGET`, **Rule Flow** = `ACCOUNT_MODIFY`

☐ Expand the `ACCOUNT_MODIFY` twisty in the bottom of the left pane (if not already expanded)

☐ Expand the **Rules Package** section in the right pane (the **+** icon beside Rules Package)




The Live Events / TARGET queue is where the connectors and adapters write all events into IGI, such as changes from a reconciliation. The ACCOUNT_MODIFY flow rules are run for each Modify Account event. For example, if one of our AD accounts was modified in AD, then next recon would sent that account to IGI as a Modify Account event.

Whilst there are two rules sitting in the Rules Package (a library of rules for this event type) only the Modify Account rule is actually run (as shown in the left pane). This rule will perform the actual account modification in IGI (and thus cannot be removed).

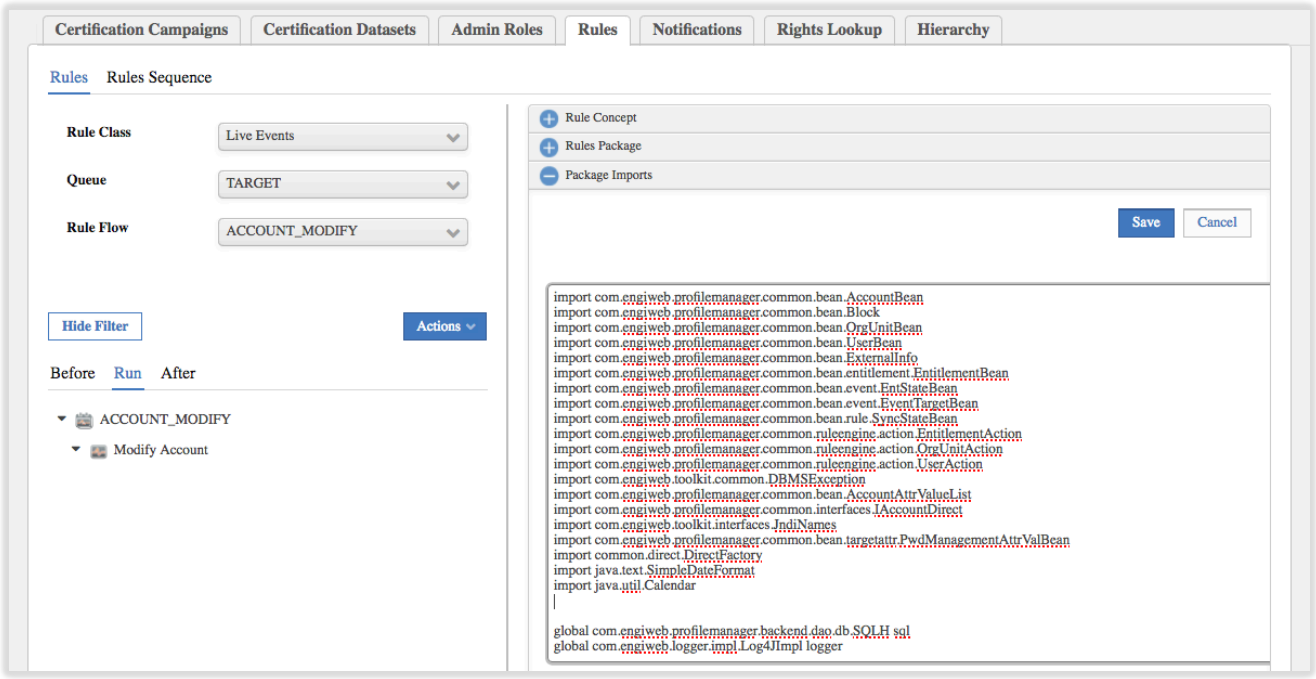
We need to add a new rule after this rule to modify the person attribute:

- ☐ In the right pane, expand the **Package Imports** section
- ☐ Have a look at the code there – it is defining all of the includes for the rules
- ☐ Add import `com.engiweb.profilemanager.common.bean.ExternalInfo` after the `import com.engiweb.profilemanager.common.bean.UserBean`
- ☐ Add the following lines after the import `common.direct.DirectFactory` and before the first `global` statement; `import java.text.SimpleDateFormat` and `import java.util.Calendar`

The resulting package imports should look like:

```
import com.engiweb.logger.impl.Log4JImpl
import com.engiweb.profilemanager.backend.dao.db.SQLH
import com.engiweb.pm.entity.BeanList
…
import com.engiweb.profilemanager.common.bean.OrgUnitBean
import com.engiweb.profilemanager.common.bean.UserBean
import com.engiweb.profilemanager.common.bean.ExternalInfo
import com.engiweb.profilemanager.common.bean.entitlement.EntitlementBean
import com.engiweb.profilemanager.common.bean.event.EntStateBean
…
import com.engiweb.toolkit.interfaces.JndiNames
import com.engiweb.profilemanager.common.bean.targetattr.PwdManagementAttrValBean
import common.direct.DirectFactory
import java.text.SimpleDateFormat
import java.util.Calendar

global com.engiweb.profilemanager.backend.dao.db.SQLH sql
global com.engiweb.logger.impl.Log4JImpl logger
```

Note, the order doesn't really matter.

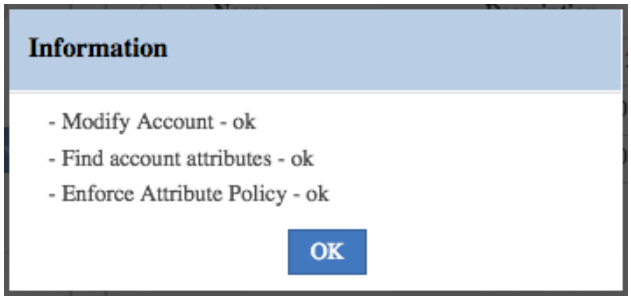



- ☐ Click **Save** and click **OK** on the Informational dialog
- ☐ Expand the **Rules Package** section
- ☐ Click **Actions > Create** to create a new rule package
- ☐ On the **Replace With…** dialog provide a rule name such as "`Enforce Attribute Policy`" and optionally a description
- ☐ In the large box, replace the "`when then`" with the following code:

```
when
    event : EventTargetBean(  )
    userBean : UserBean(  )
    orgUnitBean : OrgUnitBean(  )
    accountBean : AccountBean(  )
    accountAttrValue : AccountAttrValueList(  )
then
//
// UserERC key attribute used for fake enforce!
final String ENFORCE_KEY = "AcctLastRecon";

// If Identity not found exit
if (userBean == null || userBean.getId() == null) {
       logger.info("No Identity found for account: " +accountBean.getCode());
       return;
}

logger.info("Identity to update to push Enforcing :" + userBean.getCode());

// Get Identity ExternalInfo
ExternalInfo userExternalInfo = UserAction.findExternalInfo(sql, userBean);

// Get todays date and time
Calendar currentTime = Calendar.getInstance();
String stringDate = new SimpleDateFormat("dd-MM-yyyy
HH:mm:ss").format(currentTime.getTime());

logger.info("!!! Current Time:  " + stringDate);

// Get Current Key value
try {
       String value = (String) userExternalInfo.getAttribute(ENFORCE_KEY);
       logger.info("Previous Recon Date :" + value);
} catch (Exception ex) {
       // Skip and set 0
}

// Set the new value
userExternalInfo.setAttribute(ENFORCE_KEY, stringDate);

// Update all user attributes
UserAction.modifyUser(sql, userBean, userExternalInfo);

logger.info("Identity Updated!");
```

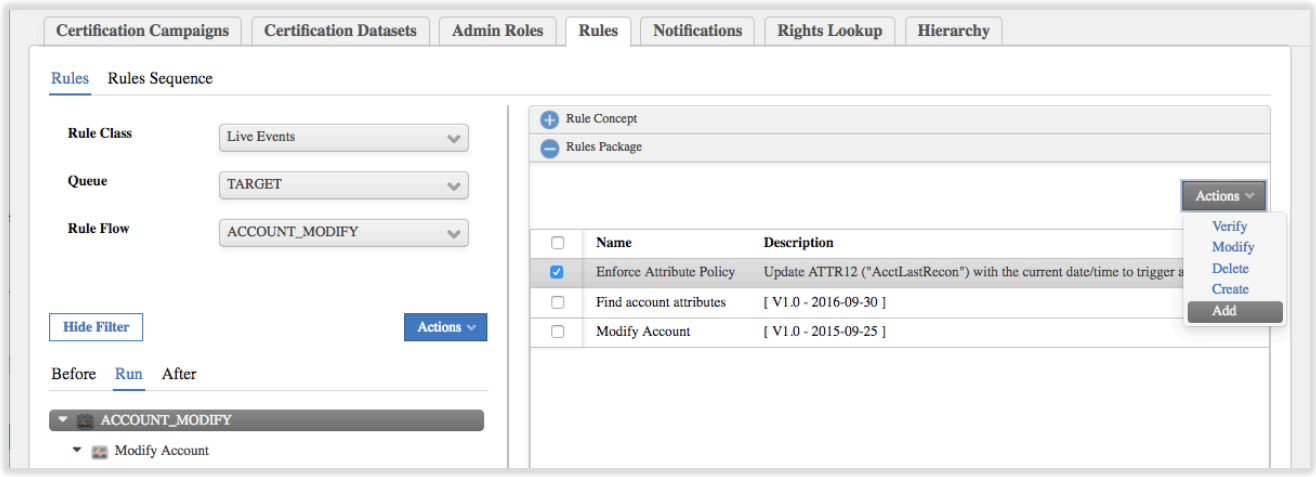
Make sure you get all of the code. We will explain what the code does in the next section.

- ☐ If you didn't give `ATTR12` the "`AcctLastRecon`" label, you need to change the highlighted code to you label
- ☐ Click **Save**
- ☐ Click **Actions > Verify** to check the code is syntactically correct and the objects/methods can be resolved

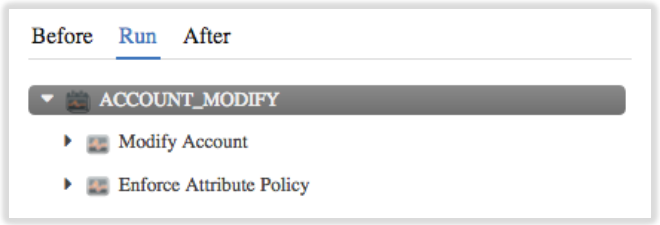



If you get errors relating to strings or quotes, it may be the copy and paste. Try pasting the rule code into notepad and then copying to the Rules editor page.

- ☐ Click **OK** on the Information dialog
- ☐ Now select the `ACCOUNT_MODIFY` in the lower left pane and the new `Enforce Attribute Policy` rule in the **Rules Package** section and click **Actions > Add**




This should have placed the new rule below the existing one.




This rule is now active and awaiting the next account modify event in the TARGET queue. This completes the IGI configuration for the two lab scenarios. The next section describes the rule we added but is included for information only.

### 3.3.5.4 Understanding the New Enforce Attribute Policy Rule

This rule will set the current system date/time into ATTR12 which will trigger IGIs re-evaluation of the attribute policy.

The first bit of code checks for the existence of a number of beans that will have been setup by the IGI event processing and store them in local beans we can use in the rule. The rule shouldn't run if all of these beans aren't present.

We have access to the event information (limited value), the user, the OU of the user, the (generic) account and the extended account attributes (in that order below).

```
when
    event : EventTargetBean(  )
    userBean : UserBean(  )
    orgUnitBean : OrgUnitBean(  )
    accountBean : AccountBean(  )
    accountAttrValue : AccountAttrValueList(  )
then

//
// UserERC key attribute used for fake enforce!
final String ENFORCE_KEY = "AcctLastRecon";
```

The ENFORCE_KEY string sets the attribute to be used. Notice that it's using the label of the attribute, not the attribute name itself.

```
// If Identity not found exit
if (userBean == null || userBean.getId() == null) {
	logger.info("No Identity found for account: " +accountBean.getCode());
	return;
}

logger.info("Identity to update to push Enforcing :" + userBean.getCode());

// Get Identity ExternalInfo
ExternalInfo userExternalInfo = UserAction.findExternalInfo(sql, userBean);
```

It then checks that the user bean has been setup by IGI and that the ID is not null. Theoretically this rule shouldn't run if there is no user (i.e. the account is unmatched or an orphan).

The last bit of code will get the external info (the extended person attributes) and store them for use.

The rest of the rule will get the current date+time from the Calendar object, format it into "dd-MM-yyyy HH:mm:ss" (e.g. 31-12-2017 12:01:01) and write it to the ENFORCE_KEY attribute specified above (in our case ATTR12 (which is AcctLastRecon). The try/catch loop is a bit pointless (copied from another example).

```
// Get todays date and time
Calendar currentTime = Calendar.getInstance();
String stringDate = new SimpleDateFormat("dd-MM-yyyy
HH:mm:ss").format(currentTime.getTime());

logger.info("!!! Current Time:  " + stringDate);

// Get Current Key value
try {
	String value = (String) userExternalInfo.getAttribute(ENFORCE_KEY);
	logger.info("Previous Recon Date :" + value);
} catch (Exception ex) {
	// Skip and set 0
}

// Set the new value
userExternalInfo.setAttribute(ENFORCE_KEY, stringDate);

// Update all user attributes
UserAction.modifyUser(sql, userBean, userExternalInfo);

logger.info("Identity Updated!");
```

This rule is ready to go and doesn't need any more configuration.

If you later need to change the rule, you need to ensure that the rules engine cache is configured to load it quickly. To check the cache settings, go into Process Designer, select the Rules Engine task and look at the Jobs associated with it. The cacheTime argument shows the cache time (default is 120mins). You can set it to a lower number if you're working on rules. You will need to stop the task, modify the setting, then restart the task for the change to be applied.

When testing, you can check the `accessgovernancecore_event_target.log` (IGI VA LMI, Manage > Custom File Management, log > iga_core folder) for the logger messages from the rule.

## 3.3.6 Running the Lab Scenarios

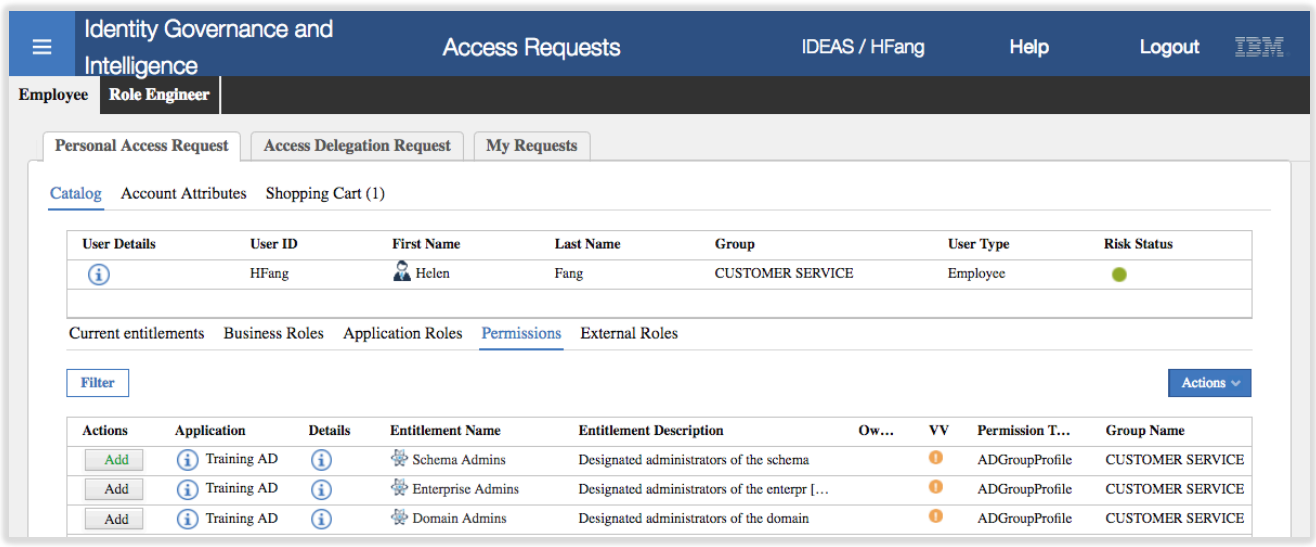
As the reconciliation was performed during the adapter setup, there is no need to rerun it (it won't show anything anyway). There are two scenarios we want to run through; Provisioning with account creation, and reconciliation with attribute policy enforcement.

### 3.3.6.1 Demonstrating Provisioning with New Account Creation

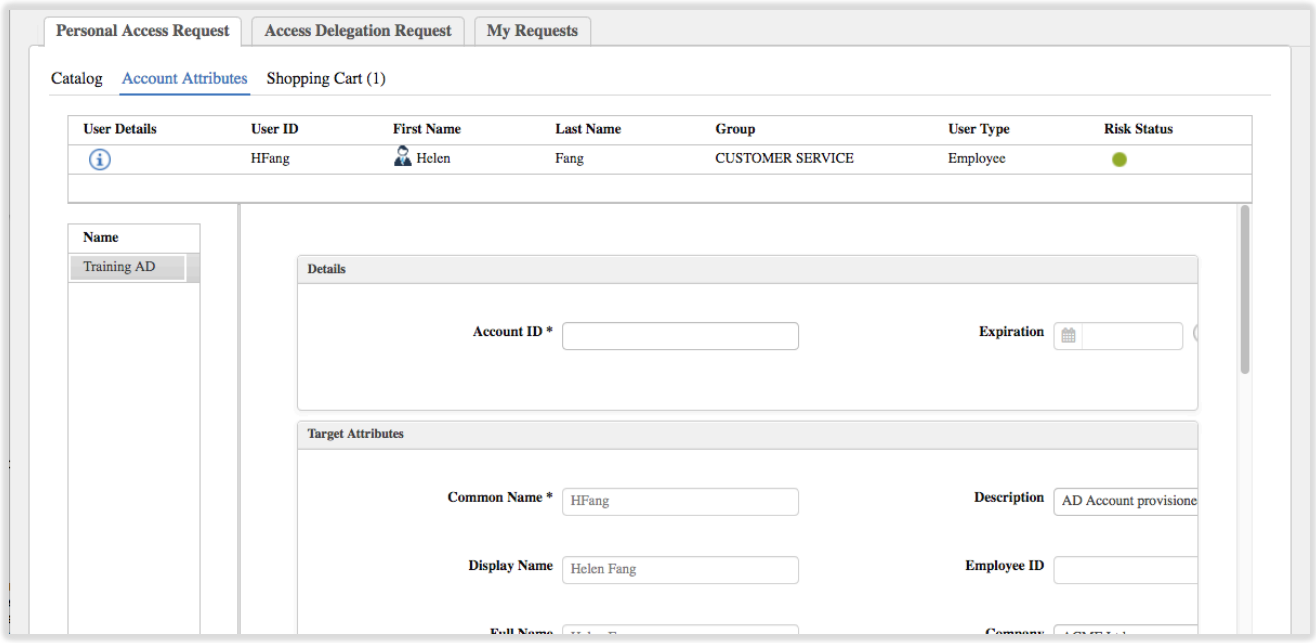
You need to identify a user who doesn't already have an AD account for this. We will use Helen Fang, but any user who doesn't have a Training AD account would work.

To run this scenario:

- ☐ Log into the **Service Center** as `Helen Fang` (`HFang`/`Passw0rd`)
- ☐ Go to **Access Requests** and note that Helen only has a QA role & JohnsonControls permission (no AD)
- ☐ Click on the **Permissions** tab and select `Training AD` as the application
- ☐ Add one of the AD permissions, such as `Schema Admins`




- ☐ Click **Next**
- ☐ On the **Account Attributes** page select the `Training AD` account



This page is shown because we enabled the “Enable Account Creation” flag to true in the workflow activity. If we had left it as false, this page would not be displayed and IGI would generate the account with attribute values in the background.

The page allows specification of the Account ID. It is a mandatory field.

> Note, there is currently a limitation in IGI such that this field cannot be pre-populated. Whilst the policy for UserID creation is in the Account configuration (in this case the ID is equal to the Ideas account id), IGI must be evaluating it later in the flow. You just have to enter a value in to proceed past this screen.

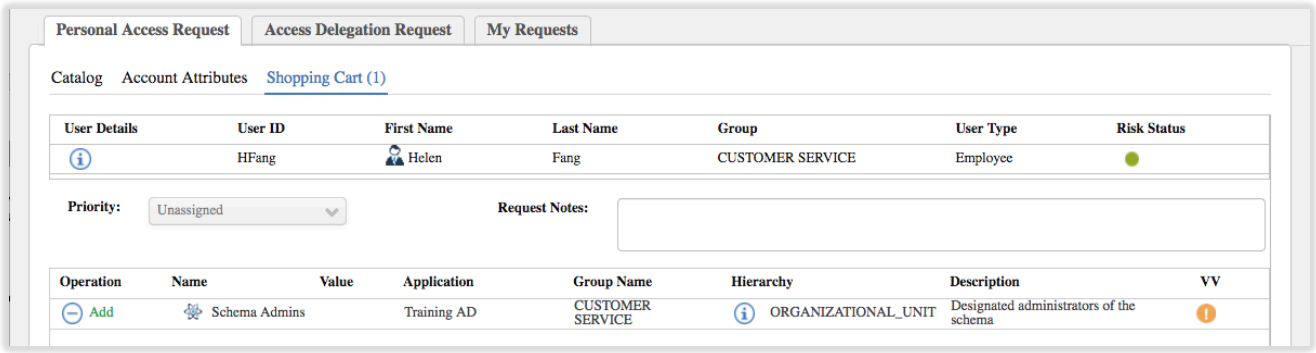
The page also shows the Target Attributes we configured above. They are in the same order as defined in the workflow activity (we didn’t change the order). Most of the attributes are read only, except for Description, Company and Department that we left editable. Note how the values have been created from the person attributes (or blank if there is no corresponding person attribute value).

Right at the bottom of the page is the Password section. This specifies the account password and must be set.

☐ Enter a **New Password** and **Confirm Password** (any password, we won’t use it)

Notice that the password policy is evaluated as you type the new password.

☐ Click **Next**




Notice that the new permission has the Visibility Violation flag set (VV orange icon). This is because we set Visibility Violation on the entitlement when we associated it with the org tree. The use of VV allows us to flag entitlements that we want to flag as possible risks (we could also create Sensitive Access risks and map them to these permissions, but that is more work – use of VV is a simpler approach).

☐ Click **Submit** and **OK** on the Request Submitted dialog

We need to approve this change as the manager:
☐ Log out and back into the Service Center as `DFox` (`Passw0rd`) – Helen’s manager
☐ Go the **Access Requests**
☐ Go to the **User Manager** tab and **Authorize Employee Request**
☐ Select the **sub-request ID** for the `Role Assign` request for Helen (it should be at the top)
☐ In the top panel on the right is an Accounts to assign label with an information icon. Click the icon.

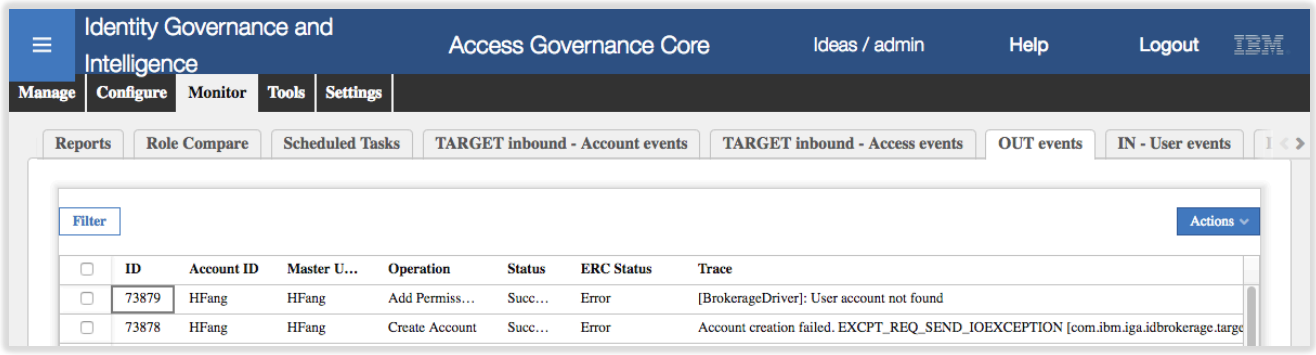
This view shows all of the account attributes for the new account. Note that they are all read-only.

☐ Click **Close**
☐ On the request pane click **Approve**
☐ Click **OK** on the Details dialog

The request has disappeared off David’s request list.

Next, we will check that the changes have been applied:
☐ Log out of the Service Center and log into the **Admin Console** (`admin`/`admin`)
☐ Go to **AGC > Monitor > OUT events**

You should see two events for HFang, a Create Account and an Add Permission event.



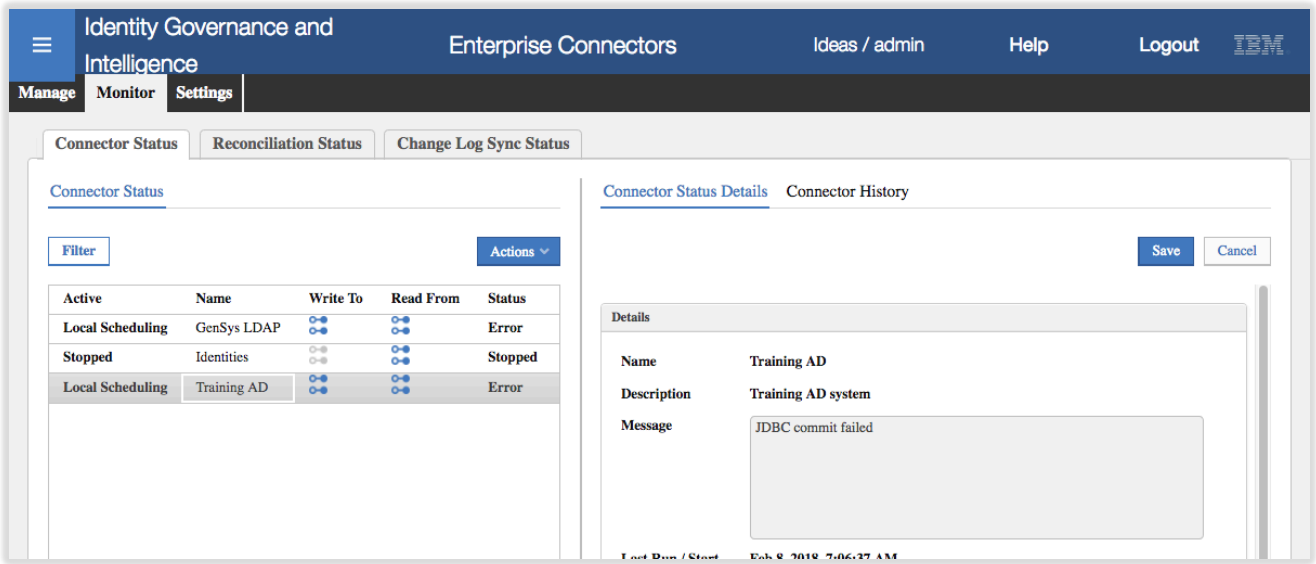

If the Status is Success, then all of the internal IGI processing of the event (including any rules) completed successfully. If not, there may be some problem with the rules.

If the ERC Status is Success, then the provisioning to AD worked. If not, there was a problem with the provisioning. That could mean a problem with the connector in the Enterprise Connectors module, a problem in the Identity Brokerage, or a problem with the adapter/agent code itself.

Obviously, you should check that the AD system is running.

The first place to check is the connector in Enterprise Connectors (sometimes in the training system when the VA has been connected, the connectors fail) and if there is an error, restart the connector.

To do this:
☐ Go to **Enterprise Connectors** and look at the **Connector Status**

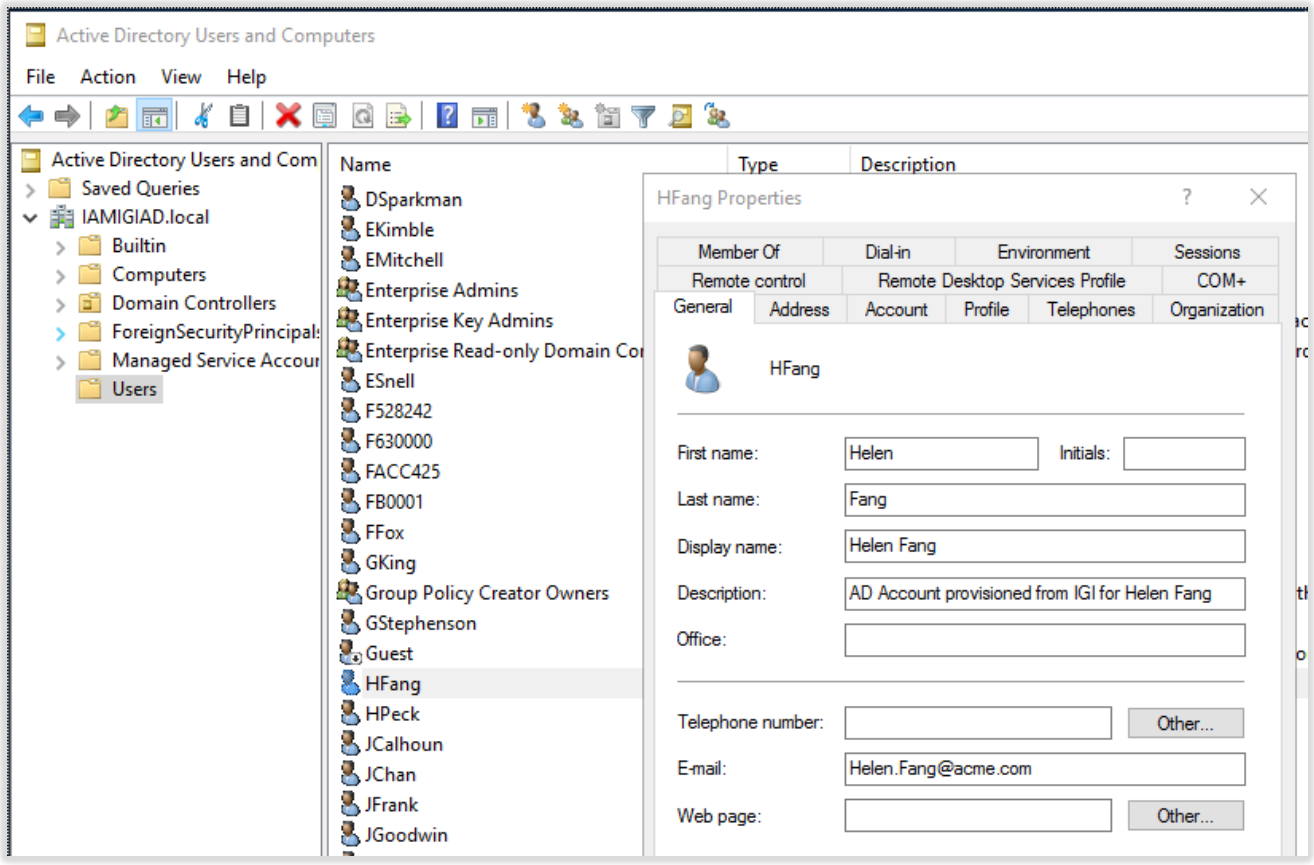



☐ If it is in error, use **Actions > Stop** and **Actions > Start** to restart, then go back to **AGC > Monitor > OUT Events**, select the two events and click **Actions > Re-execute**

If this is not the problem, you may need to check the connector logs, the identity brokerage logs and the agent logs. We don't cover debugging in this lab.

When the events are successfully processed we can go into AD and check the results.

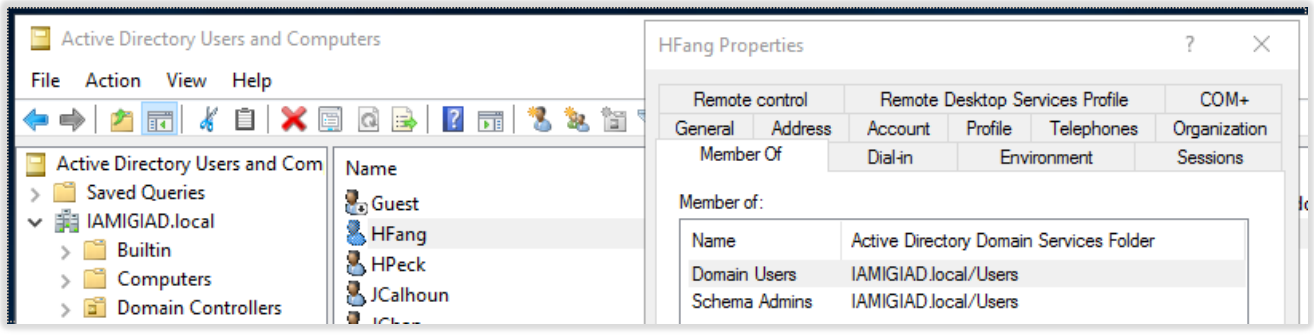
☐ Log into the **AD** system (`NetworkAdmin`/`Passw0rd`)
☐ Open **Active Directory User and Computers** (it's in the task bar/system tray)
☐ Expand the **Users** folder and look for `Helen Fang` (will be towards the bottom)
☐ Double click `Helen` to see the account details




You can see the account details based on the mapping we defined in IGI.

☐ Click on the **Member Of** tab

You should see Helen is a member of Domain Users and Schema Admins.




This completes this scenario. We have demonstrated that we can create an AD account just by selecting an entitlement in IGI and control the account attribute values based on user attribute values, fixed strings or combinations. We can expose or hide the mapping of these in the access request workflow. These values are carried down through the connector and agent and applied to Active Directory.

In the next section we will look at the reconciliation with attribute policy enforcement use case.

### 3.3.6.2 Demonstrating Reconciliation with Attribute Policy Enforcement

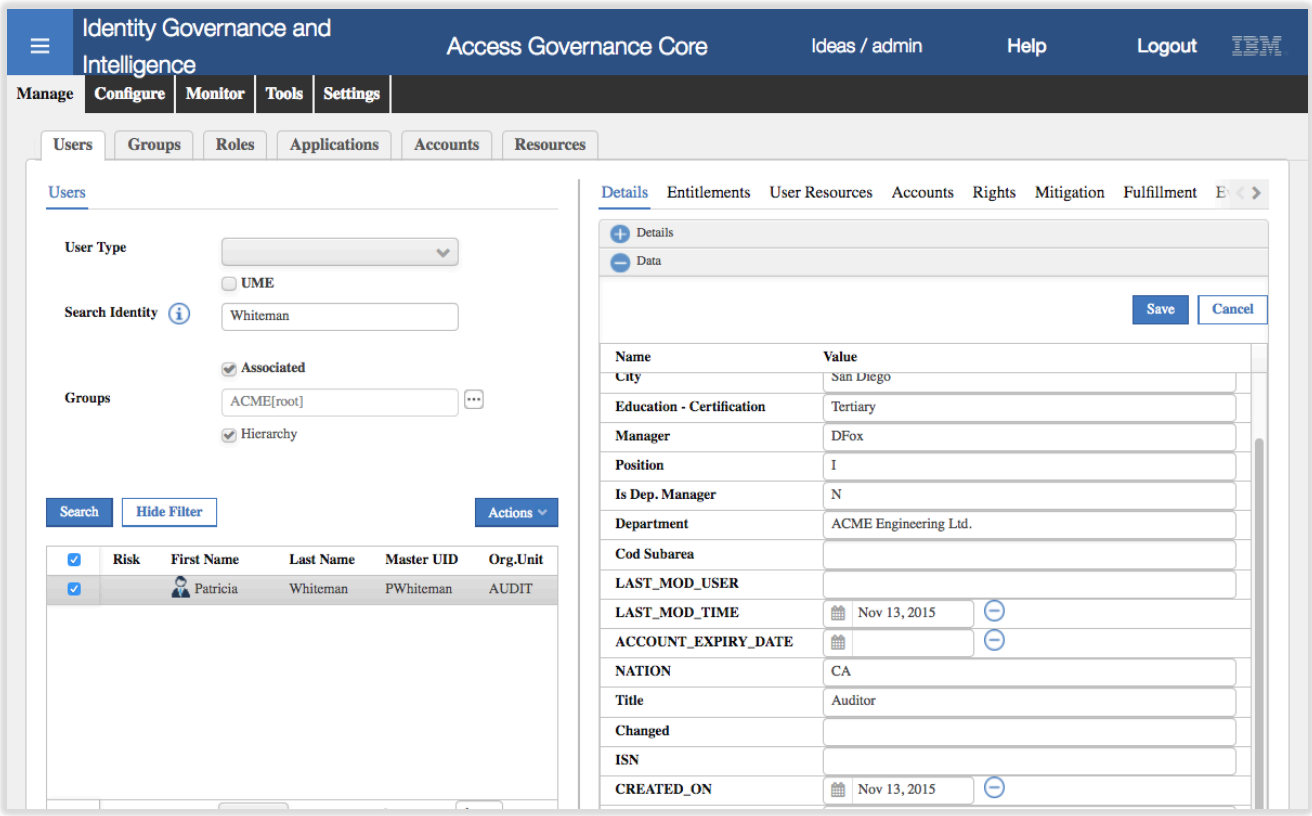
This flow is a bit more complex than the previous. We want to show that attribute policy enforcement works when a reconciliation is performed.

For this you will need a person who has an account on the AD (perhaps the one from above, like HFang, or one of the ones from the first recon, like PWhiteman/Passw0rd). We will use PWhiteman.

You will also need to select one of the attributes you flagged for enforcement in the target attribute configuration. We will use Title (ATTR10).

The first step is to check/set the person attribute in IGI:

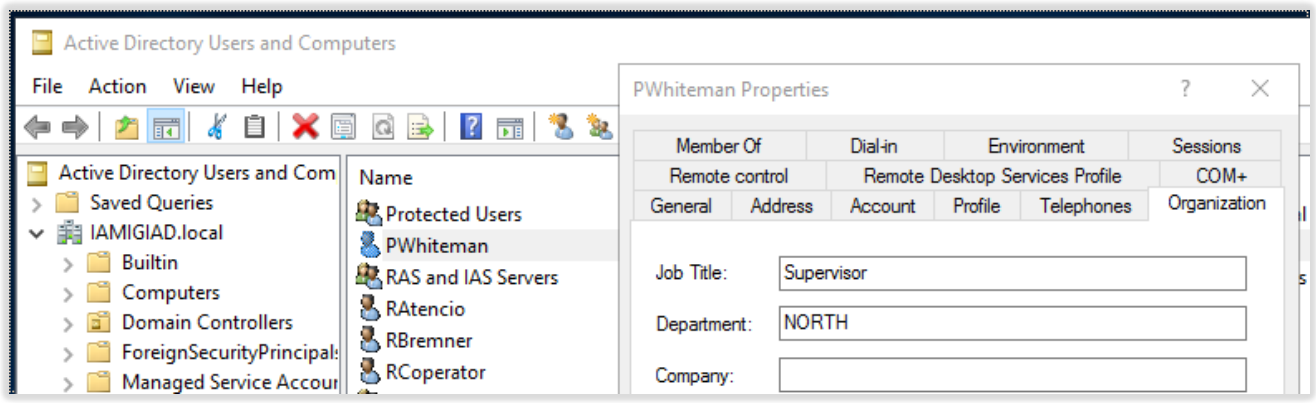
- ☐ Log into the **Admin Console** (`admin`/`admin`) and go to **AGC** (default view is Manage > Users which we want)
- ☐ Find and select `Patricia Whiteman`
- ☐ Expand the **Data** section in the right pane to see the extended attributes
- ☐ Find the `Title` attribute




You can see Patricia has a title of Auditor. So, in this scenario we're assuming this value has been set from a HR Feed. In this training environment you will see (next step) that Patricia's AD account does not have that Title value as the account was created and reconciled before we setup the target attribute mapping. However, we want this value to stand no matter what value is set in AD.

Next, we need change the title in AD

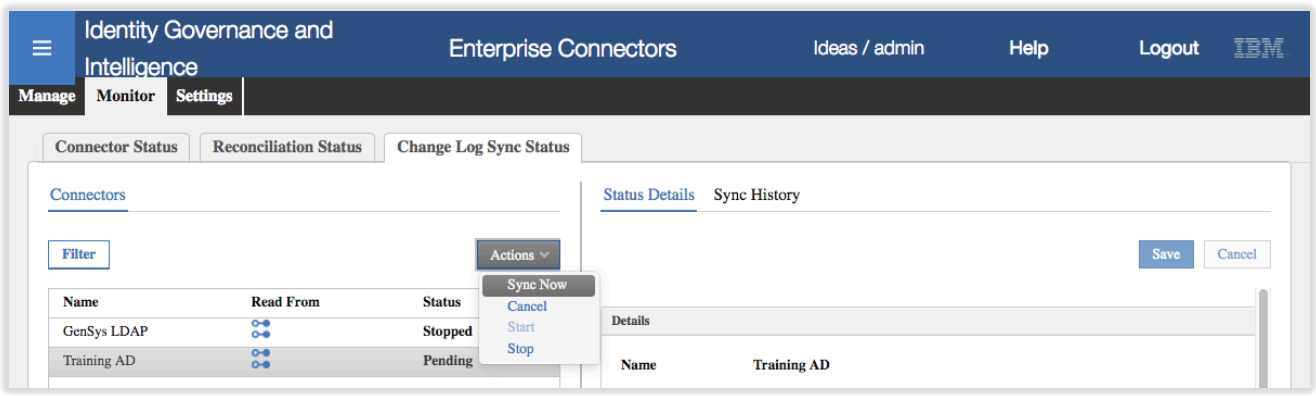
- ☐ Log into the **AD server** (`NetworkAdmin`/`Passw0rd`)
- ☐ Open **Active Directory User and Computers** (it's in the task bar/system tray)
- ☐ Expand the **Users** folder and look for `Patricia Whiteman` (will be towards the bottom)
- ☐ Double click `Patricia` to see the account details
- ☐ Go to the **Organization** tab and notice that the **Job Title** is blank
- ☐ Enter a new title, like "`Supervisor`" and click **OK**



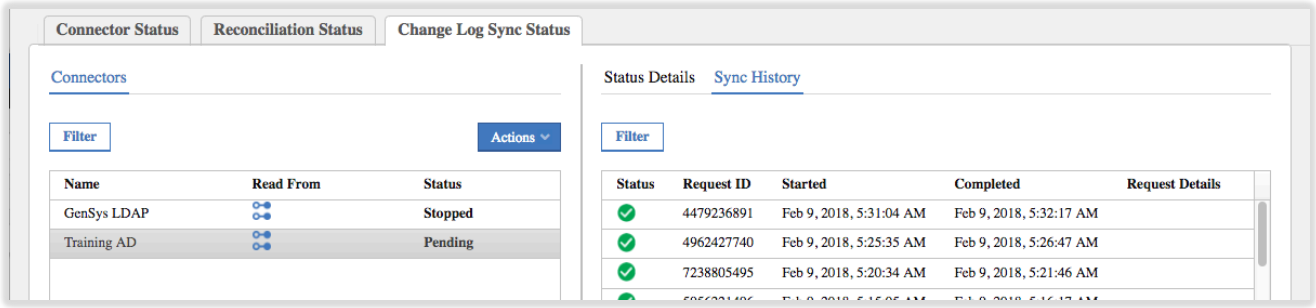

We now need to either wait for a reconciliation cycle to run, or force. You may be lucky to have the Change Log Sync and Read-From Channel sync occur soon after the change.

We will force it:

☐ Log into the **IGI Admin Console** (`admin`/`admin`) and go to **Enterprise Connectors**
☐ Go to **Monitor > Change Log Sync Status**
☐ Select the `Training AD` connector and click **Actions > Sync Now**



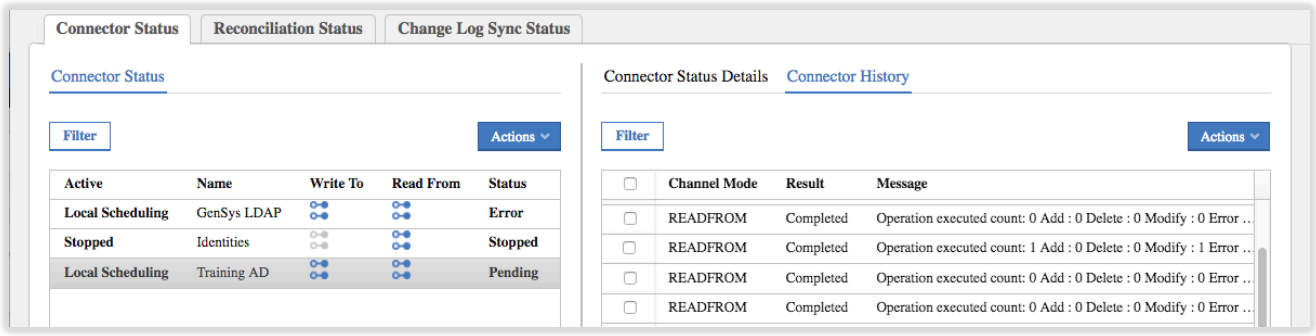

☐ Go to the **Sync History** tab and wait for the Sync to complete




This means the agent has run a recon and the Identity Brokerage has written and changes to the delta table.

☐ Go to the **Connector Status** tab and look at the connector history view for `Training AD`

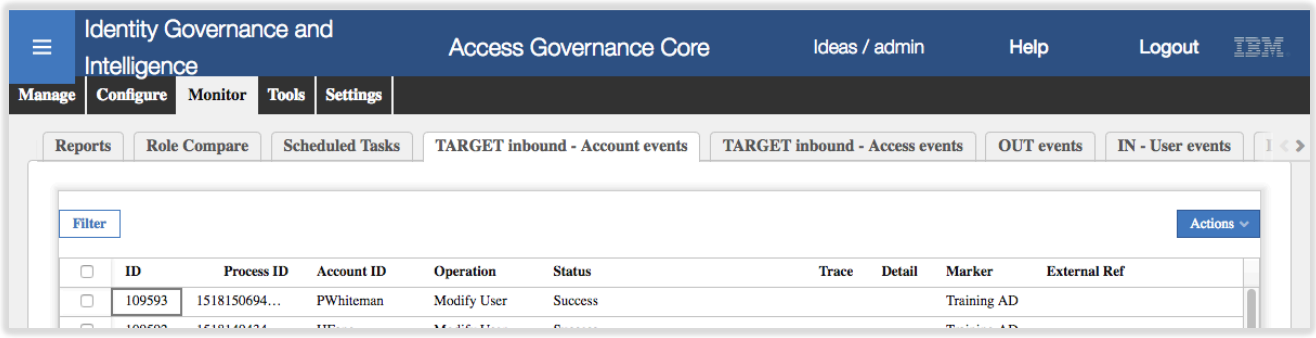
When we configured the connector earlier we set the changelog sync to evert five minutes, whereas the Read From sync for every minute. There should be an event in the history that is newer than the changelog sync and shows one (1) Add operation.




If not, just wait a minute. Note you can force an immediate sync by stopping and starting the connector.

We can now see if the incoming account modify event has triggered the rule and expectant behavior:

☐ Go to **AGC > Monitor > TARGET inbound – Account events**



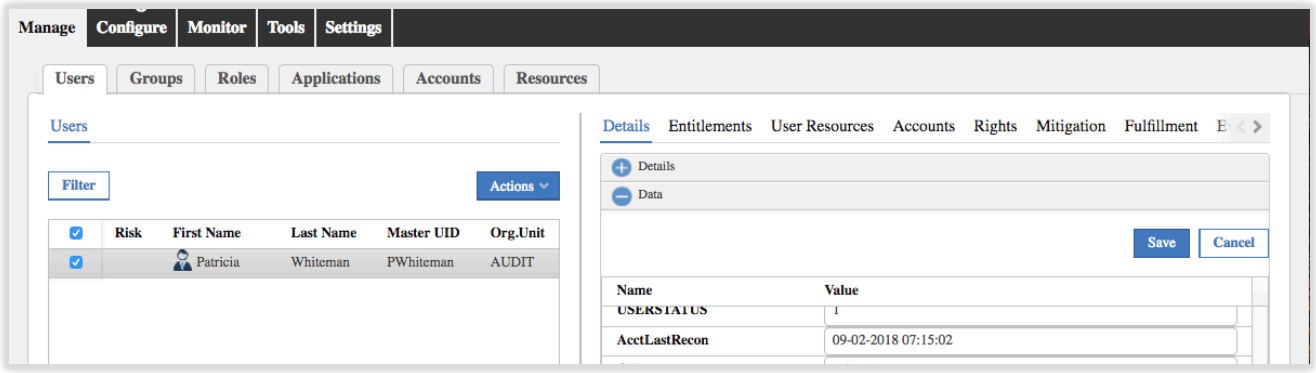

There should be a recent Modify User (actually a Modify Account) event for PWhiteman.

If the Status is success, that means that all of the Modify Account rules have run successfully. To confirm:

☐ Go to **AGC > Manage > Users**

☐ Find and select `Patricia Whiteman` and expand the **Data** section in the right pane

☐ Look for the `AcctLastRecon` attribute

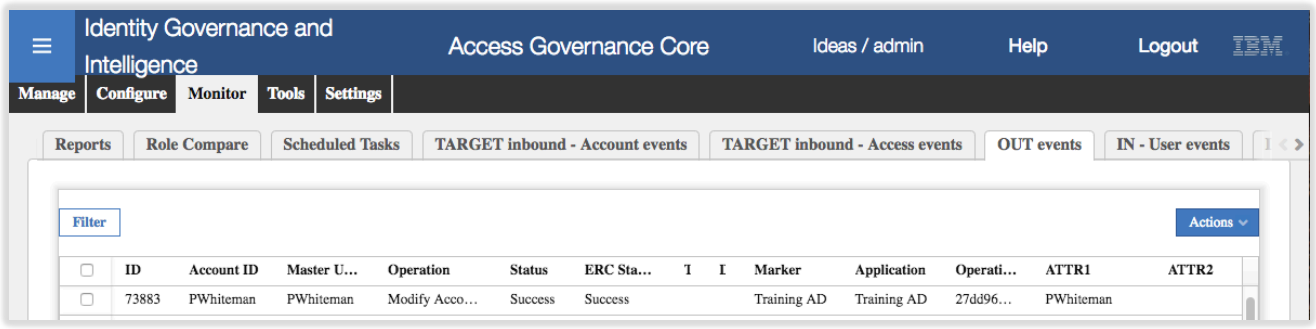



If the attribute is set to a recent date+time, then the rule has executed successfully.

To see if the attribute enforcement worked:

☐ Go back to **AGC > Monitor > OUT events**

If the enforcement worked there should be a Modify Account event present for PWhiteman.




The account title should be changed to Auditor in AD.

> There seems to be a bug in the current training system. Sometimes the attribute re-evaluation works as expected, sometimes you see the person attribute modified (ATTR12/AcctLastRecon) but no re-evaluation occurs nor is there a change sent back to the AD system.

This completes Lab 1.

# 4 Lab 2 – A User Dept. Move Triggers a Continuous Campaign

This lab is built around the requirement for a manager to review any access after a person changes their department. It will involve configuration and use of:

- Continuous campaigns
- User modify event rules
- User modify workflow and ARM in the Service Center

This Lab only needs the IGI VAand Data Server VMs. It does not need the Windows Server VM.

## 4.1 Overview of Scenario

This scenario is summarized in the following table. This is just for your information.

| | |
|---|---|
| ***Summary*** | Configuring a continuous campaign and dataset, and java rules for feeding the dataset (based on the Rules guide) and then run some user moves (with contractor management in the service center) |
| ***Requirement statement*** | Users belong to departments, defined in the organizational structure.<br><br>There are default/enforced permissions associated with each department, along with some optional ones that users can request.<br><br>When a user moves department their access for that new department must be reviewed by their manager. |
| ***Demonstration requirements*** | Must be able to demonstrate:<br>• User move operation<br>• Manager reviews new access in cert campaign |
| ***Implementation notes*** | This will require configuration of the following:<br><br>• **Certification** - a continuous user entitlement campaign and dataset setup with manager as the reviewer.<br>• **Event Rule** - rule to add an entry to the cont. campaign dataset hooked into the User Move event<br>• **Workflow** - user modify workflow enabled with some user attributes, including the org unit<br><br>Assume: GenSys LDAP is the target, admin roles are there (user manager etc.) |
| ***Skills required*** | Students should have a good grasp of the following:<br>1. Cert campaigns and datasets (B100 trg module)<br>2. Rules (Rules Guide with cont. campaign examples and D2** trg modules)<br>3. Workflow (B300/301/B330 trg modules) |

You should be able to use the demo/training data in the training image to test this, such as DFox (manager) and PWhiteman (user) both with Passw0rd in the Service Center. The administrator is admin with password admin.

## 4.2 Lab2 – Detailed Lab Instructions

The following instructions will walk you through the lab setup and execution.

The lab will configure a number of items in IGI to support the following flow:

1. A user manager logs into the Service Center and accesses the User Modify workflow
2. They select the user and change their department
3. This triggers a User Move operation in IGI
4. In response to this operation a custom rule collects all of the entitlements and writes them to a continuous campaign dataset
5. The user manager logs into the Service Center and sees an outstanding Campaign activity for the user change and reviews the changed access.

This involves configuring:

- A certification dataset and campaign
- A user virtual view and workflow process and activities
- A Move User rule

The steps for these are covered in the following section, followed by steps to test the flow.

### 4.2.1 Configuring the Certification Dataset and Campaign

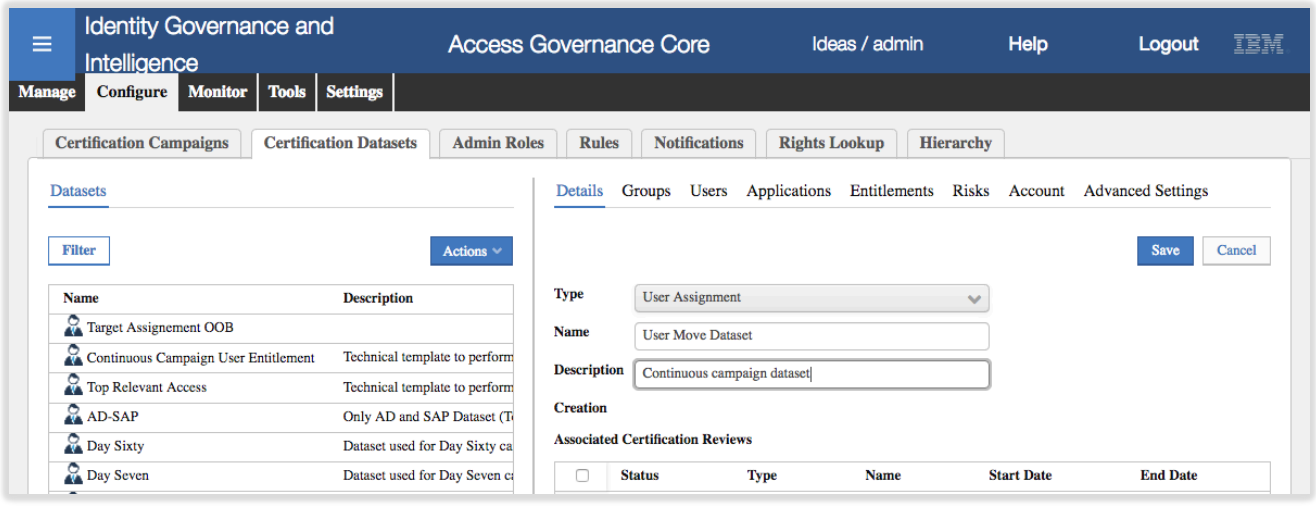
The following steps describe creating the Campaign Dataset and Certification Campaign. Both are fairly standard IGI configurations, so the following sections won't go into a lot of detail.

#### 4.2.1.1 Create Campaign Dataset

☐ Create a new certification dataset in **Access Governance Core** (**Configure > Certification Dataset**). The key properties to be specified:

- Type: "`User Assignment`"
- Name: this can be anything, but will be used in a rule later
- Description: optional

An example is shown below.




☐ Don't forget to **Save** the dataset

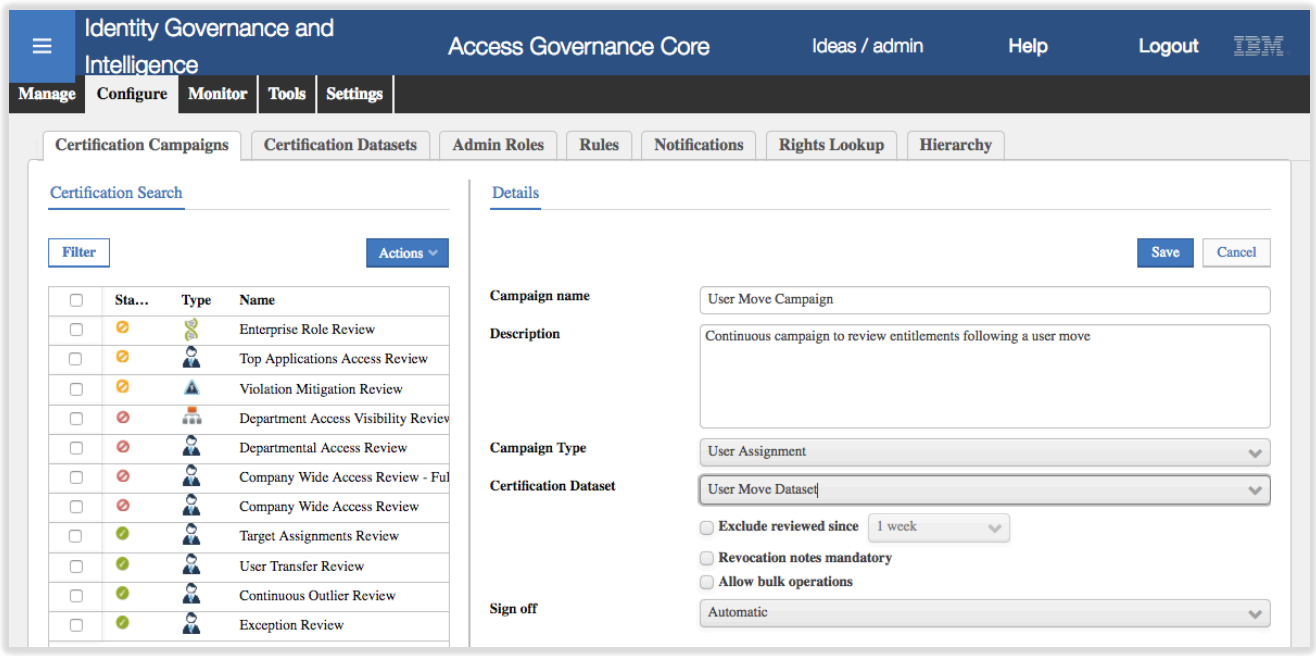
You do not need to specify any of the normal whitelists/blacklists (e.g. Groups, Users, Applications).

The dataset is complete.

### 4.2.1.2 Create Certification Campaign

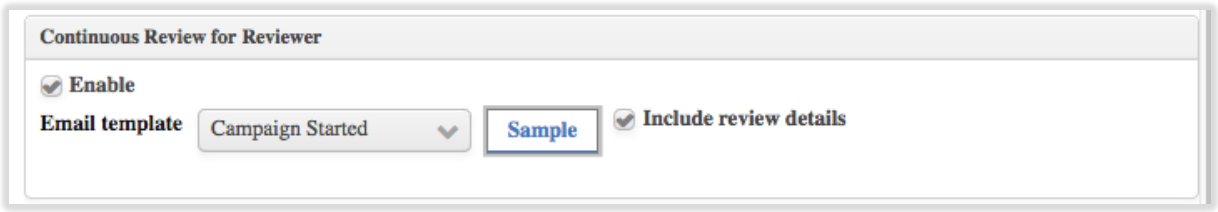
☐ Create a new certification campaign in **Access Governance Core** (**Configure > Certification Campaigns**). Unless specified, you can leave the settings as default. The details to be specified are:
- Campaign name: This can be anything
- Description: Optional
- Campaign Type: "`User Assignment`"
- Certification Dataset: the name for the dataset from the previous step

An example is shown below.




☐ **Save** the campaign
☐ On the **Supervisors** tab, assign a **Supervisor** (`Myriam Brewer` will be the only one available) and **Save**
☐ On the **Reviewers** tab, set:
- The **Scope** as `User Hierarchy, Managers`
- The **Default Reviewer** as `Shirley Chang`

☐ Leave everything else as is and **Save** the reviewers
☐ On the **Fulfillment** tab, set to `Physical deletion` and **Grace Period** to `zero days`, and **Save**
☐ On the **Scheduling** tab, set the **Duration** to `Continuous` (very important!) and **Save**
☐ We are not using email notification in this lab, but for completeness, on the **Notifications** tab enable the `Continuous Review for Reviewer` notification, select the `Campaign Started` template and the **Include review details** checkbox.




☐ **Save** the campaign
☐ Launch the campaign (**Actions > Launch**)

There is no need to View Configuration. The campaign and dataset are now ready to be used in the scenario.

## 4.2.2 Configuring the User Virtual View and User Modify Workflow

To allow the manager to modify one of their people in the service center, we need to define the appropriate workflow in the process designer. The generate activity in the process will require a user virtual view defined.

### 4.2.2.1 Define a User Virtual View

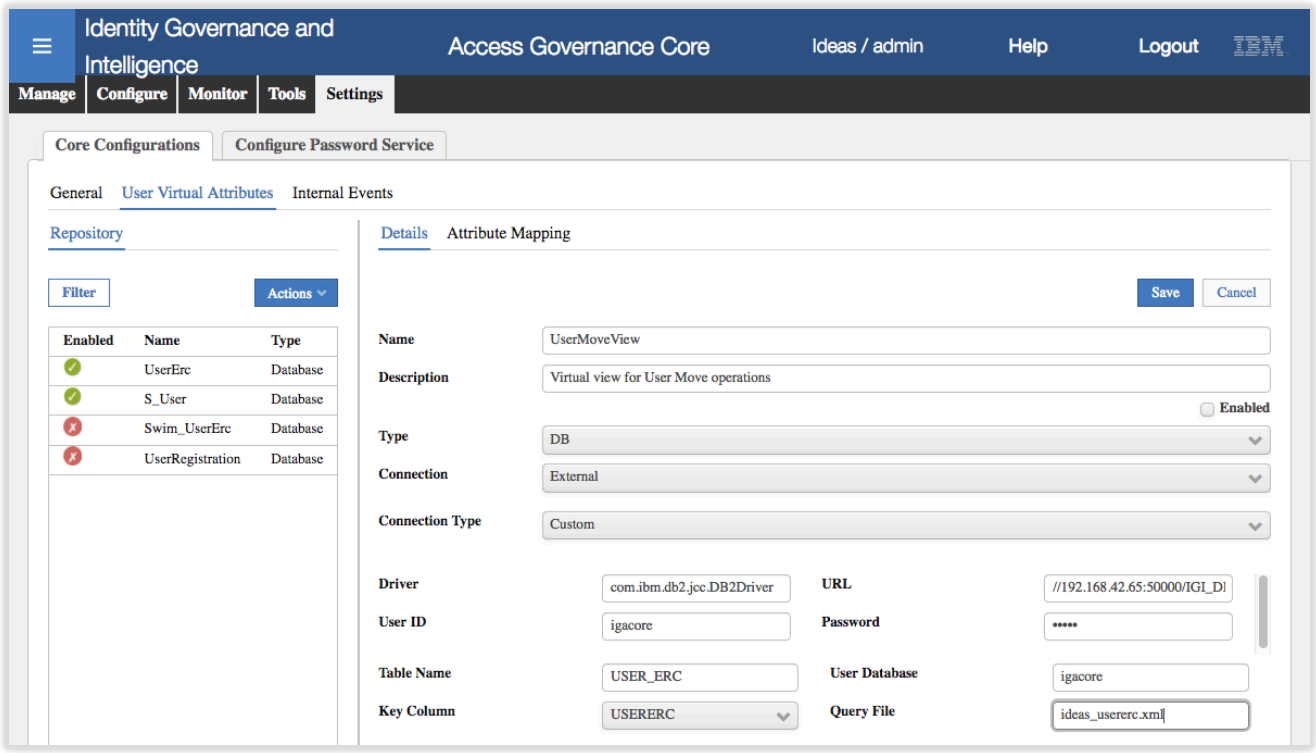
A user virtual view defines the user attributes available to user workflows.

☐ Go to **Access Governance Core > Settings > Core Configurations > User Virtual Attributes**

☐ Create a new repository (user virtual view) with the following Details:

- Name: specify a unique name (no spaces)
- Description: optional
- Type: DB
- Connection: External
- Connection Type: Custom
- Driver: com.ibm.db2.jcc.DB2Driver
- URL: jdbc:db2://192.168.42.65:50000/IGI_DB
- User ID: igacore
- Password: ideas
- Table Name: USER_ERC
- User Database: igacore
- Key Column: USERERC
- Query File: ideas_usererc.xml

It should look similar to the following:




☐ Do NOT select **Enabled** (the virtual view must be disabled)
☐ **Save** the virtual view

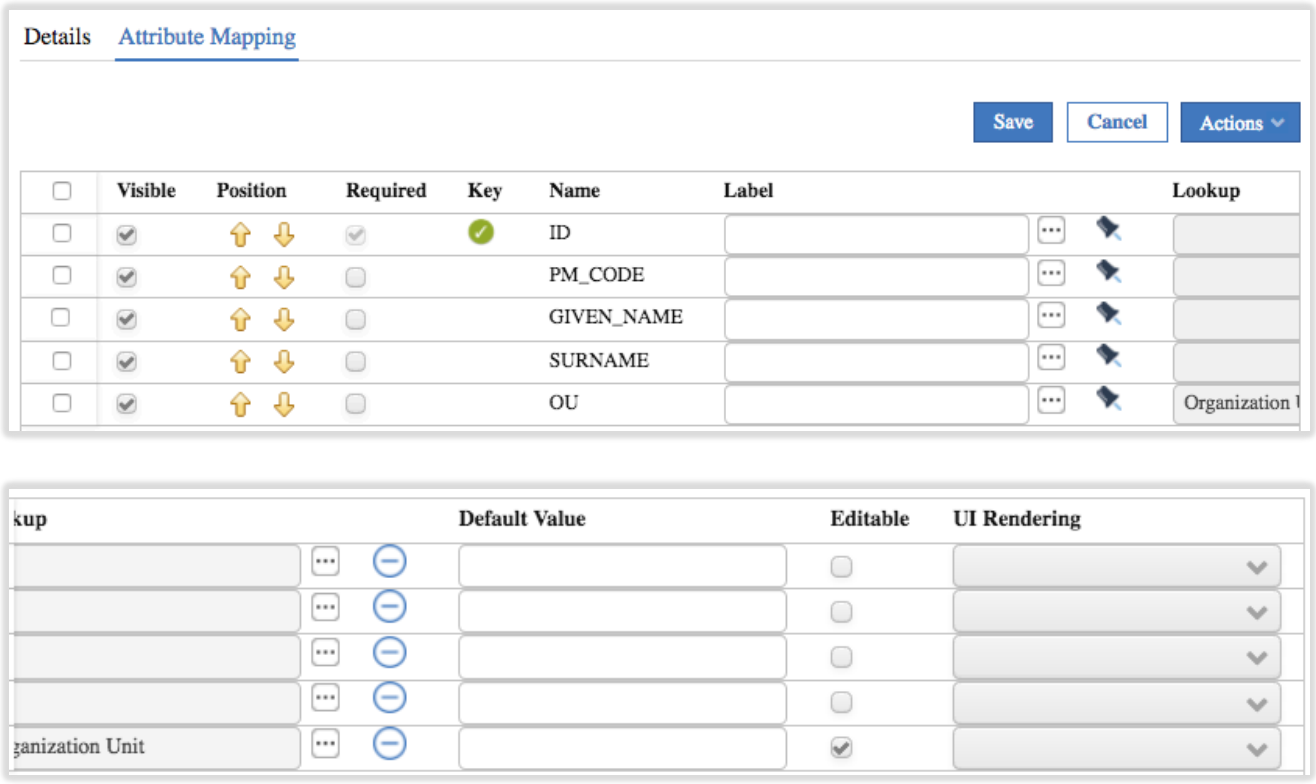
Next, we will set the attributes that can be managed in the user workflow.

☐ Go to the **Attribute Mapping** tab and select **Actions > Add**

☐ Select the `ID, PM_CODE, OU, GIVEN_NAME` and `SURNAME` attributes and click **OK**

With the attributes in the **Attribute Mapping** view:
☐ Make all the attributes **visible**
☐ Use the position arrows to sort the fields into the following **order**: `ID, PM_CODE, GIVEN_NAME, SURNAME and OU` last
☐ Click the ellipses button ([…]) beside the **Label** field for each of `PM_CODE, GIVEN_NAME, SURNAME and OU` and set the English values to "`UserID`", "`First Name`", "`Surname`", and "`Department`" (note that these do not show in the Attribute Mapping display)
☐ Click the ellipses button ([…]) beside the **Lookup** field for the `OU` attribute and select `Internal` and `Organization Unit` for the **Lookup Options**
☐ Make the `OU` the only **Editable** field
☐ **Save**

The attribute list should look like the following:




We can now define the workflow.

### 4.2.2.2 Define a User Move Workflow

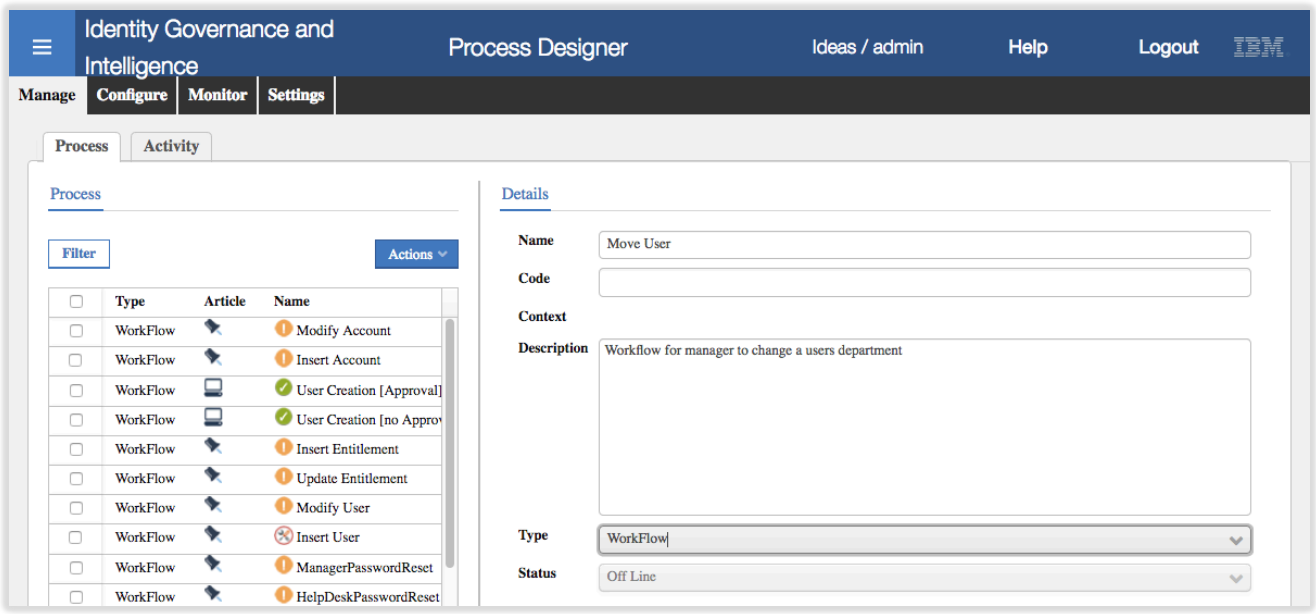
We will create a new workflow for the user move, involving just the generate step (User Manager) and an execute step (Operator).

☐ Go to **Process Designer > Manage > Process**

Notice that there is already a Modify User workflow. We will ignore that and create a new one.

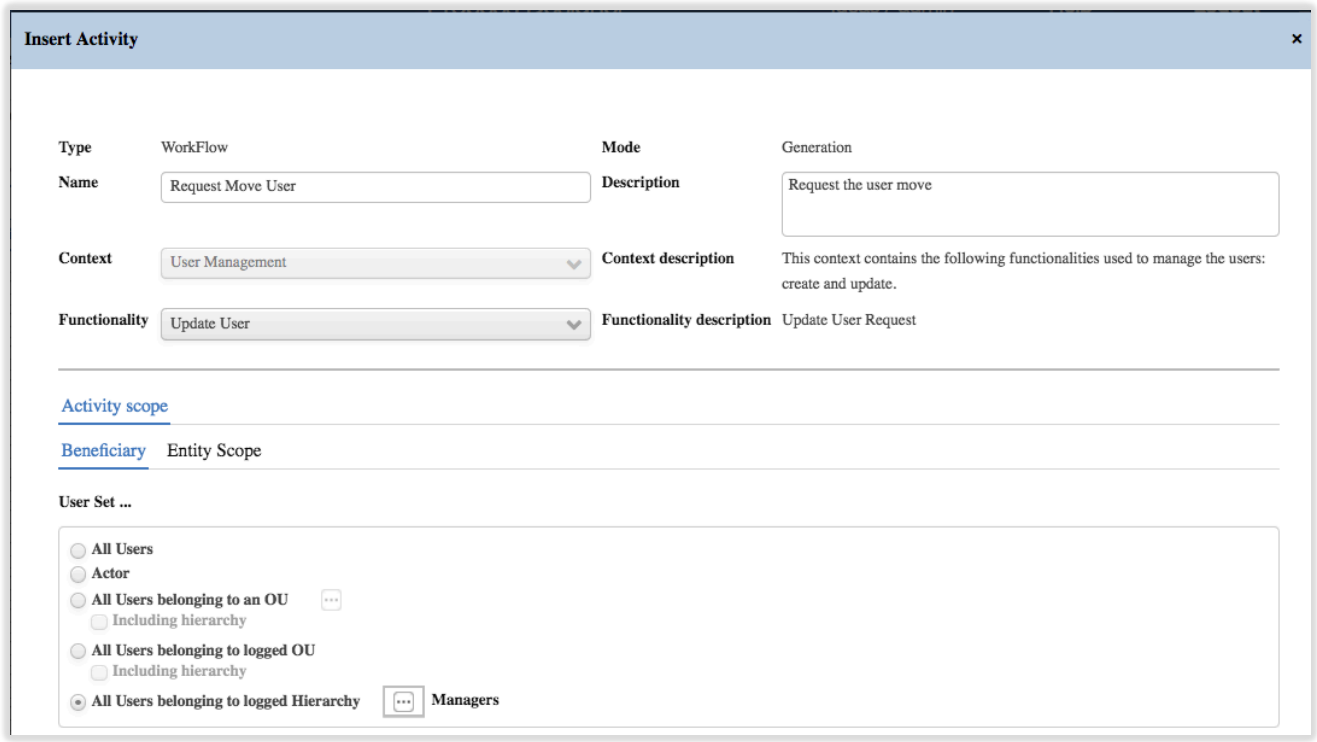
☐ Create a new workflow process (**Actions > Add**) with the following settings:
- Name: Give it a unique name (like Move User)
- Code: leave blank
- Description: optional
- Type: Workflow
- Status: leave offline for now

This should look similar to the following:

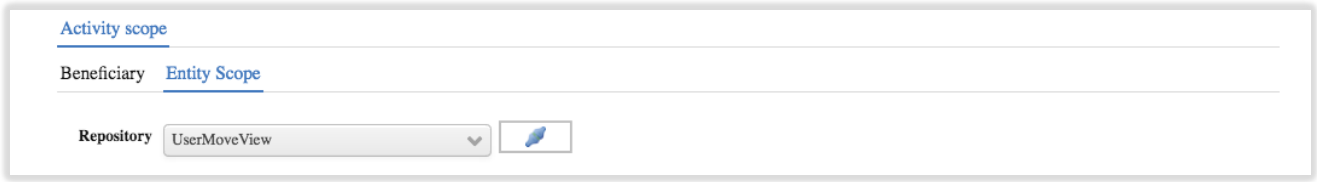



☐ Click **Next**

☐ On the **Configuration** tab, click the **Generation** icon (boxes and plus sign icon)
☐ Select (click) the icon that is now in the workspace
☐ On the **Activity** dialog, select a **Context** of `User Management`
☐ Select the `Modify User` activity and select **Create**
☐ On the **Insert Activity** dialog
- Give it a Name (such as "`Request Move User`") and Description
- Select Update User as the Functionality
- In the Activity Scope > Beneficiary section select `All Users Belonging to logged Hierarchy` of `Managers`




- ☐ Go to the **Entity Scope** tab and click the **Filter** button
- ☐ Change the repository to the user virtual view you created above (e.g. `UserMoveView`) and click to link icon to the right



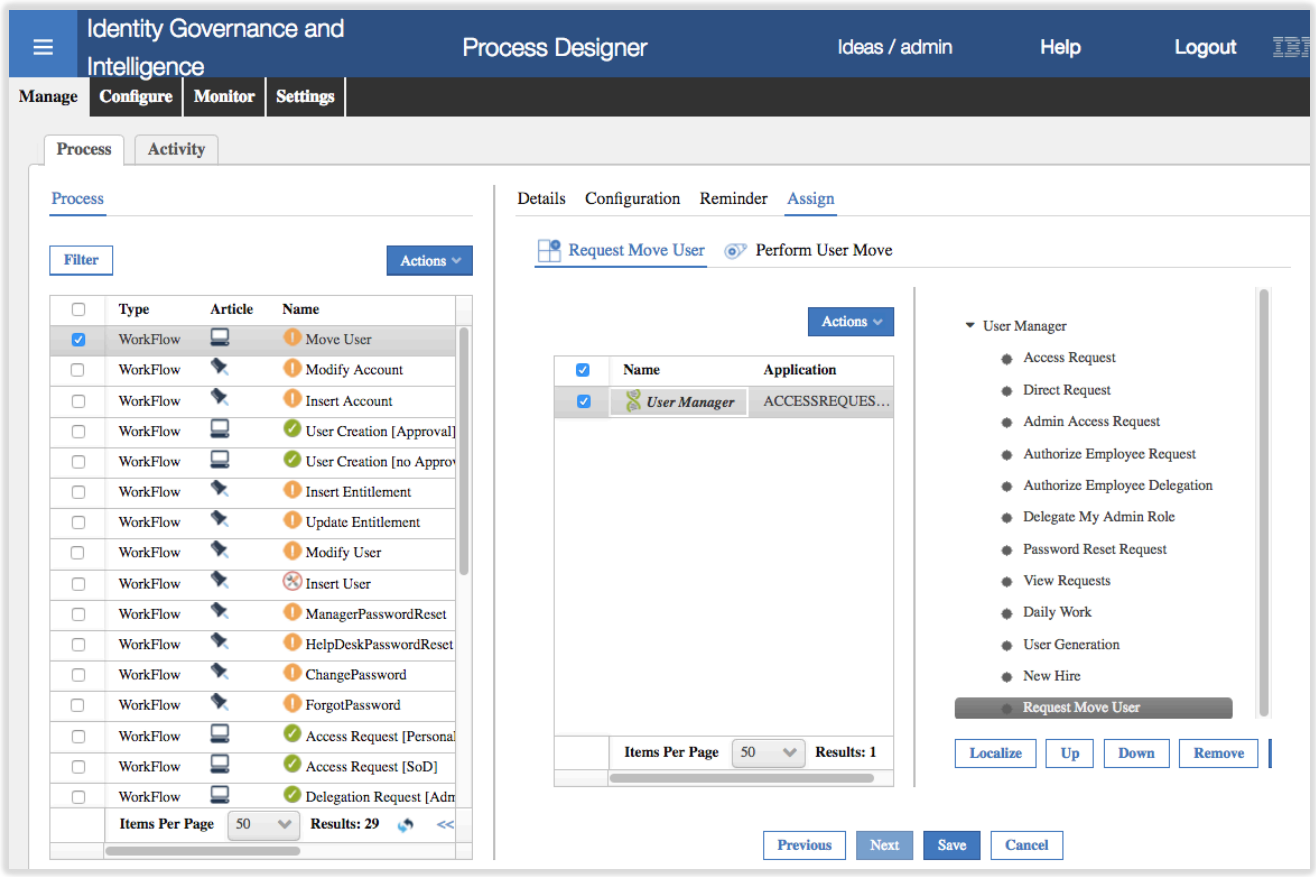

- ☐ Click **Hide Filter** to see all the attributes
- ☐ We don't need to change anything, so click **OK**

- ☐ Click on the **Execution** icon (pulley wheel icon) to create an **Execution** activity in the workspace
- ☐ Click the activity and on the **Activity** dialog, select `Exe User Modify` and click **Create**
- ☐ Give it a **Name**, **Description** and **Functionality** of `Execute Update User`
- ☐ As there is nothing else to set, click **OK**

- ☐ Click **Next**
- ☐ Ignore the **Reminder** settings and click Next

- ☐ On the **Assign** tab assign `User Manager` to the generate activity and `Operator` to the execution activity
- ☐ Click **Save** to save the workflow process
- ☐ Go back to the assignments and check the menu settings

The Assign tab for the generation activity should look similar to the following.

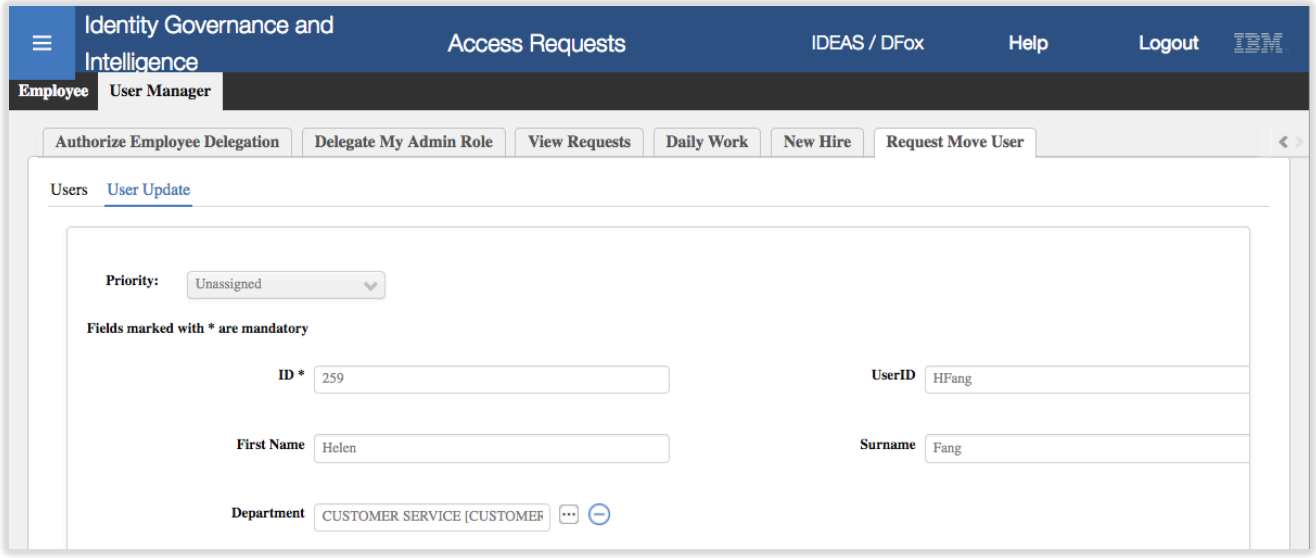



☐ Set the workflow process status to **On Line** and **Save**.

You can check this process by the following steps:

☐ Logging into the **Service Center** (as `DFox/Passw0rd`),
☐ Go to **Access Requests** and select the **User Manager** tab
☐ Tab to the right and select the **Request Move User** tab
☐ Select a user, say `Helen Fang`, and click **Next**

You should see a User Update form with the five attributes we specified in the user virtual view; ID, PM_CODE, GIVEN_NAME, SURNAME and OU. They should have the labels we set in the virtual view.



The only field that can be changed is the OU. You can click on the ellipses button ([…]) and select a different OU. We will not do this now. Just logout from the Service Center without submitting the change.

We have defined the workflow to change a user's OU and the campaign to be triggered on the move. The last step is to code the User Move event rule to write to the campaign dataset when a user is moved.

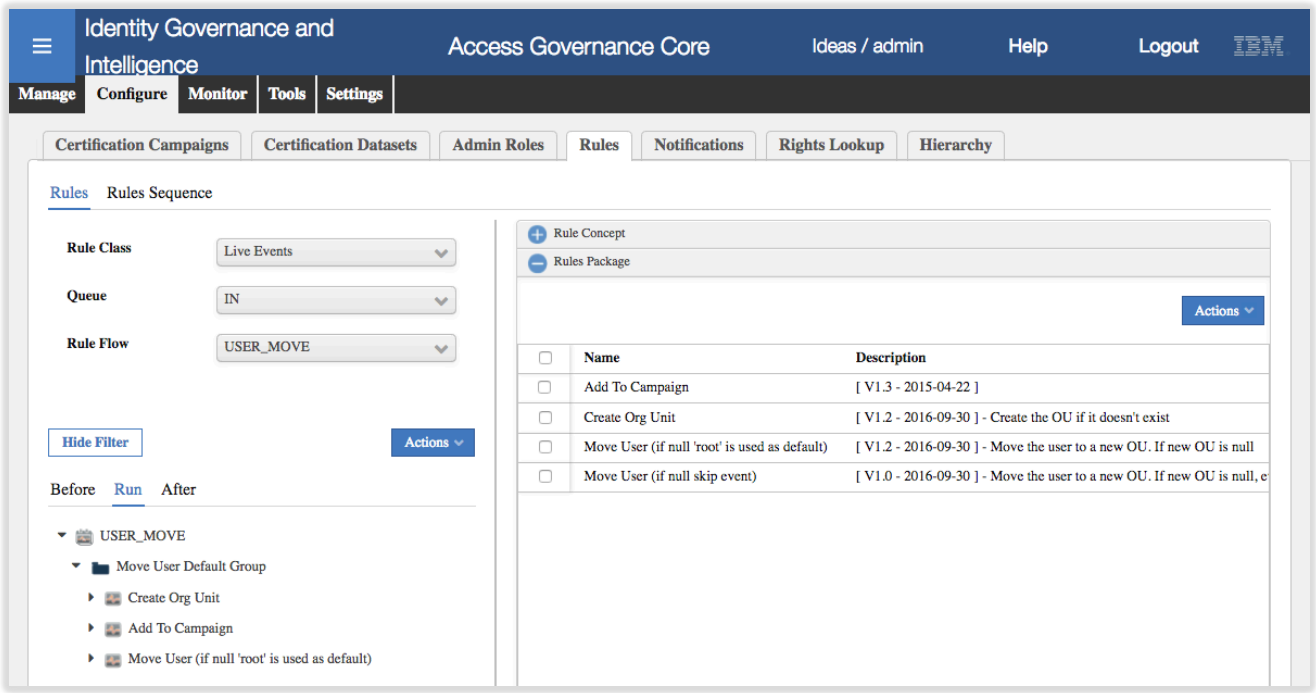
## 4.2.3 Configuring the Move User Rule

There is already an existing rule for the Move User to trigger a continuous campaign. We will modify that to use our campaign dataset. We will also need to change the rule engine task for the IN queue to reduce the rule cache time down so the changes are applied immediately.

### 4.2.3.1 Modify User Move Ruleflow

- ☐ Go to **ACG > Configure > Rules**
- ☐ Select **Rule Class** = `Live Events`, **Queue** = `IN`, **Rule Flow** = `USER_MOVE`
- ☐ In the bottom pane expand `Move User to Default Group`
- ☐ Then expand the **Rule Package** in the right pane

Your view should look similar to the following.




It's showing that there are already three rules run when a USER_MOVE event is processed; a Create Org Unit rule (to create the new org unit if it doesn't exist), an Add to Campaign rule (which we will modify) and a Move User (if null 'root' is used as default) to actually perform the move.

- ☐ In the **Rules Package** section in the right pane, select the `Add to Campaign` rule and **Actions > Modify** to edit the rule.

This is one of the supplied rules. We will briefly explore the logic before modifying it.

```
when
    userBean : UserBean(  )
    orgUnitBean : OrgUnitBean(  )
then
// [ V1.3 - 2015-04-22 ]

  // Templatename --> DefaultEmptyTemplate included in User Transfer Campaign

  String tName = "DefaultEmptyTemplate";
```

The first part is the standard when/then clause of the Drools rules engine. It will run when there is a UserBean (i.e. object with the user details) and an orgUnitBean (object with the org unit details). The first line of logic will set the name of the campaign dataset. Note that in the rules and EJB implementation, the term “template” is used to refer to a campaign dataset.

This is the bit of code we will change for our new dataset.

```
  TemplateBean templateBean = new TemplateBean();
  templateBean.setName(tName);
  TemplateDAO templateDAO = new TemplateDAO(logger);
  templateDAO.setDAO(sql);

  BeanList blTemplateBean = templateDAO.find(templateBean, new Paging(4));
  if (blTemplateBean.size()==0) {
    throw new Exception("Template does not exists!");
  }
  templateBean = (TemplateBean) blTemplateBean.get(0);
```

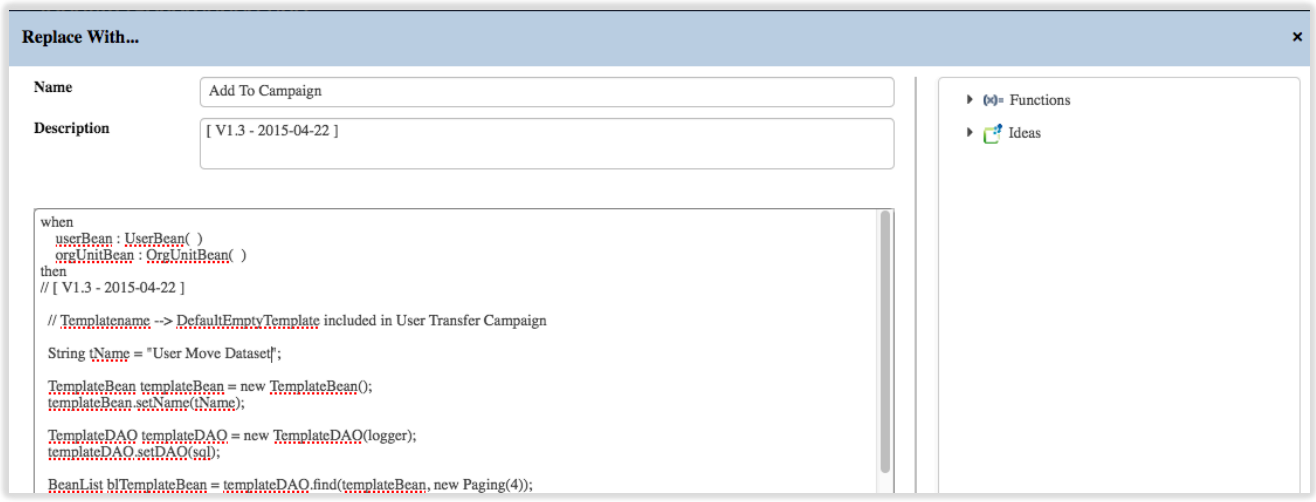
The next section of code will check to see if the dataset (template) exists and if it doesn’t it will throw an exception with the message “Template does not exists!”.

Note that this rule uses the older Data Access Object (DAO) implementation which is being deprecated in preference to the published Direct methods and Actions.

```
  BeanList entitlements = UserAction.findJobRoles(sql, userBean);

  BeanList listBean = new BeanList();
  for (int i = 0; i < entitlements.size(); i++) {
    AbstractBean[] element = new AbstractBean[2];
    element[0] = userBean;
    element[1] = (EntitlementBean) entitlements.get(i);
    listBean.add(element);
  }

  if (listBean.size() > 0) {
    templateDAO.addEntity(listBean, AttestationRes.TEMPLATE_ENTITY_USERENT, templateBean,
AttestationTypes.PERSON_ENTITLEMENT.getValue());
  }
```

The last bit of code will get the list of entitlements for the user and write them into the certification dataset.

☐ With the code open in the editor (“**Replace With…**” dialog), change the name of the template to your template name (in the example it is “`User Move Dataset`”).




☐ Click **Save** to close

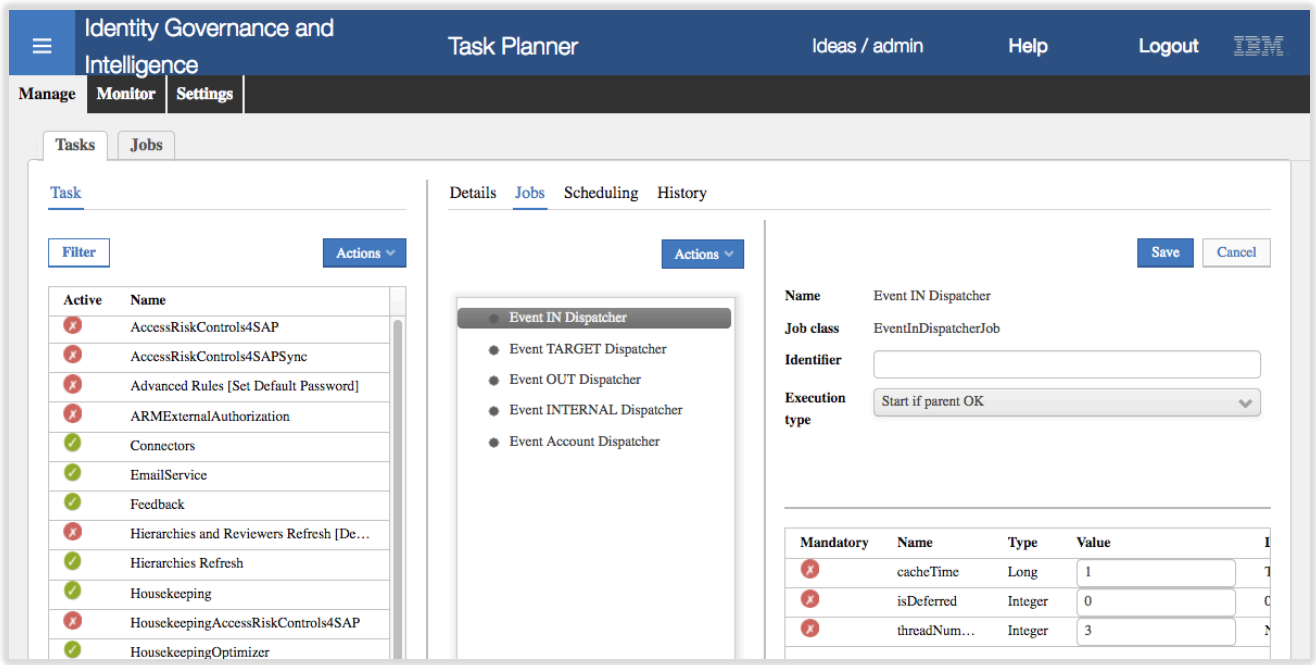
▌We could have created a new rule and copied all of the code over to it, but this will do for the lab.

We now need to go fix the rules engine cache setting.

### 4.2.3.2 Set Rules Engine Cache

☐ Go to the **Task Planner** module
☐ Select the `RuleEngine` job and stop it (**Actions > Stop**)
☐ Go to the **Jobs** tab in the right pane
☐ Select the `Event IN Dispatcher` job and find the `cacheTime` setting
☐ Change it from `120` (mins) to `1`
☐ **Save** the change

The job and task should look like the following.




☐ **Start** the Job

This will mean that the rule cache will be emptied every minute. This is useful for testing where you want your changes to be compiled and available almost immediately. In production the cache setting of 120 minutes means rules will be compiled and loaded once every two hours, which provides better performance.

We are now ready to test the flow end-to-end.

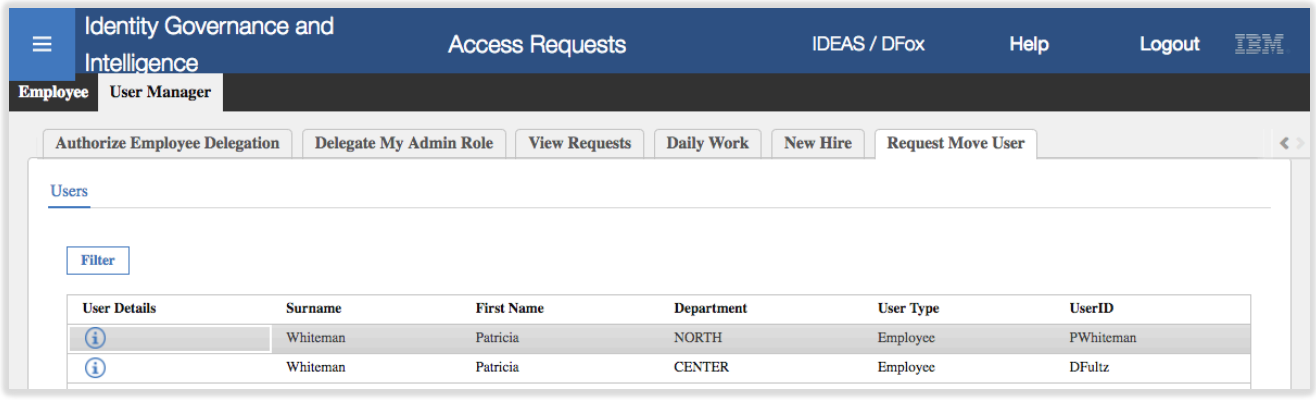
## 4.2.4 Testing

There are two parts to testing this; perform the user move, then check the recert campaign.

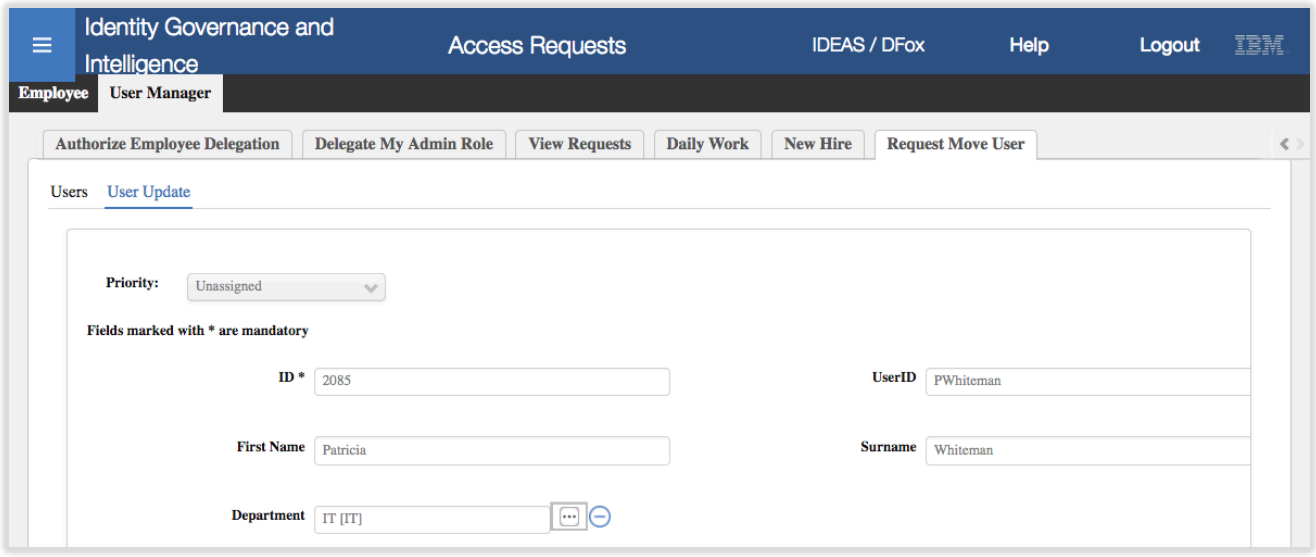
### 4.2.4.1 Testing the User Move

Log into the Service Center as DFox

☐ Go the **Access Requests**, select the **User Manager** tab and the **Request Move User** sub-tab
☐ Find and select `Patricia Whiteman` (careful, there are two, we want the one in NORTH)




☐ Click **Next**
☐ On the **User Update** form, check the fields and then use the ellipses button ([…]) beside **department** to change to `IT` (under `CORPORATE`)

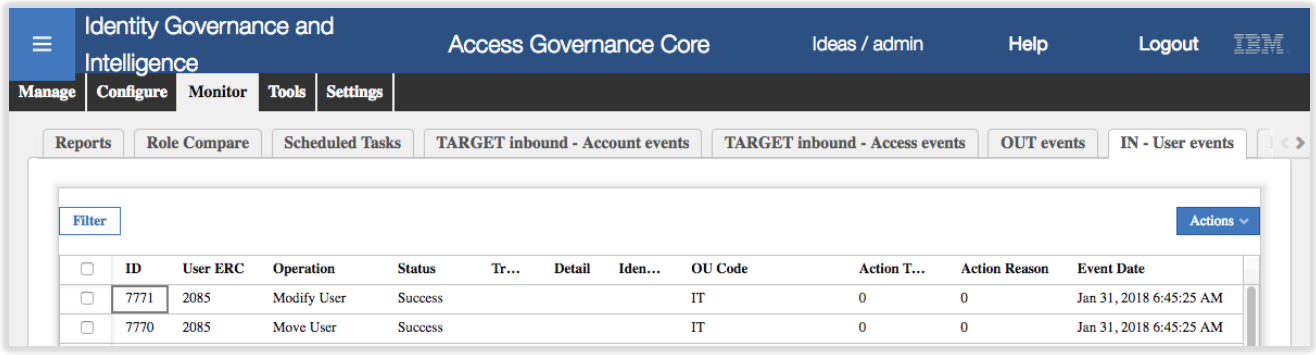



☐ Click **Submit**


☐ Click **OK** on the Request Submitted dialog

You can check on the request as follows
☐ Log into the **Admin Console** (`admin`/`admin`)
☐ Go to **Access Governance Core > Monitor > IN – User Events**

You should see two new events, a Move User event and a Modify User event. Unfortunately, neither shows the user details.




Both should have a status of Success.

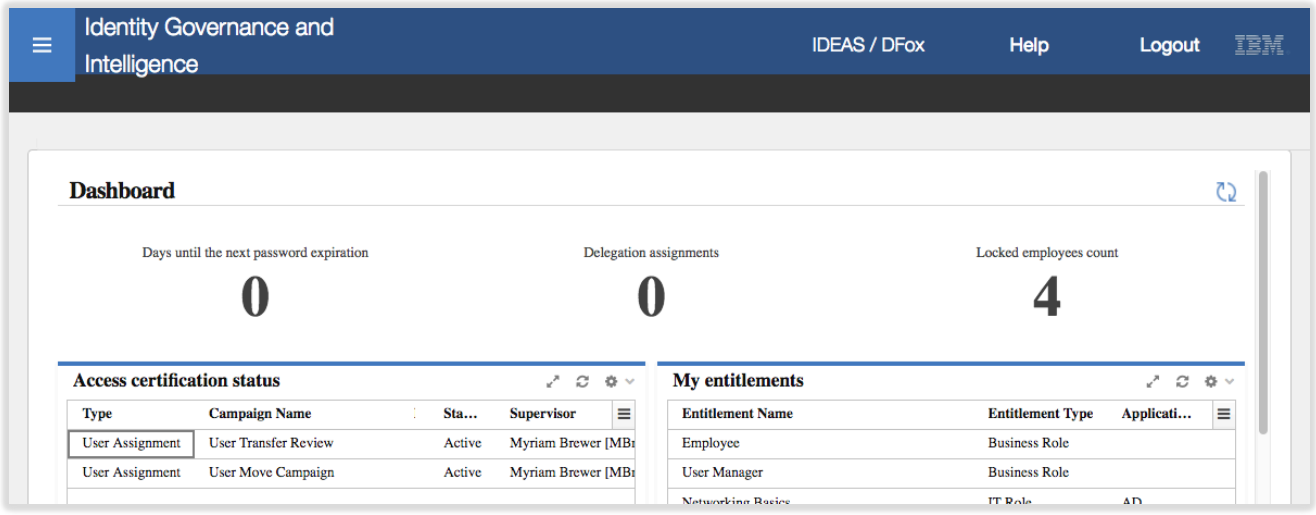
If they are still Unprocessed, refresh the screen. If they remain unprocessed for some time (a few minutes) you can check that the time between the VA and data server is in synch (see the time drift problem in the Lab Setup Guide).

For this and other issues, see the later section on debugging issues.

### 4.2.4.2 Testing the Certification Campaign

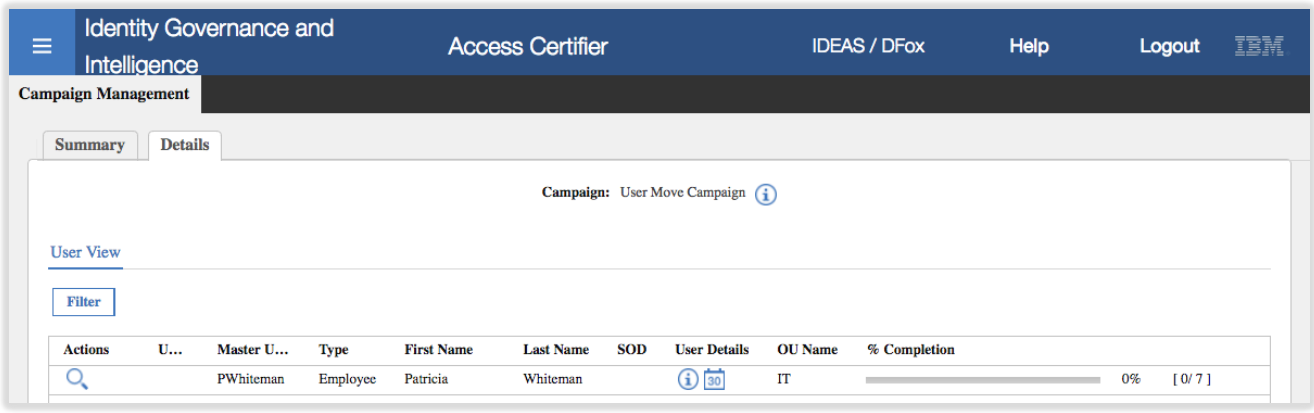
☐ Log back into the **Service Center** as `DFox`

On the dashboard you should see an Access certification status section listing your new campaign.



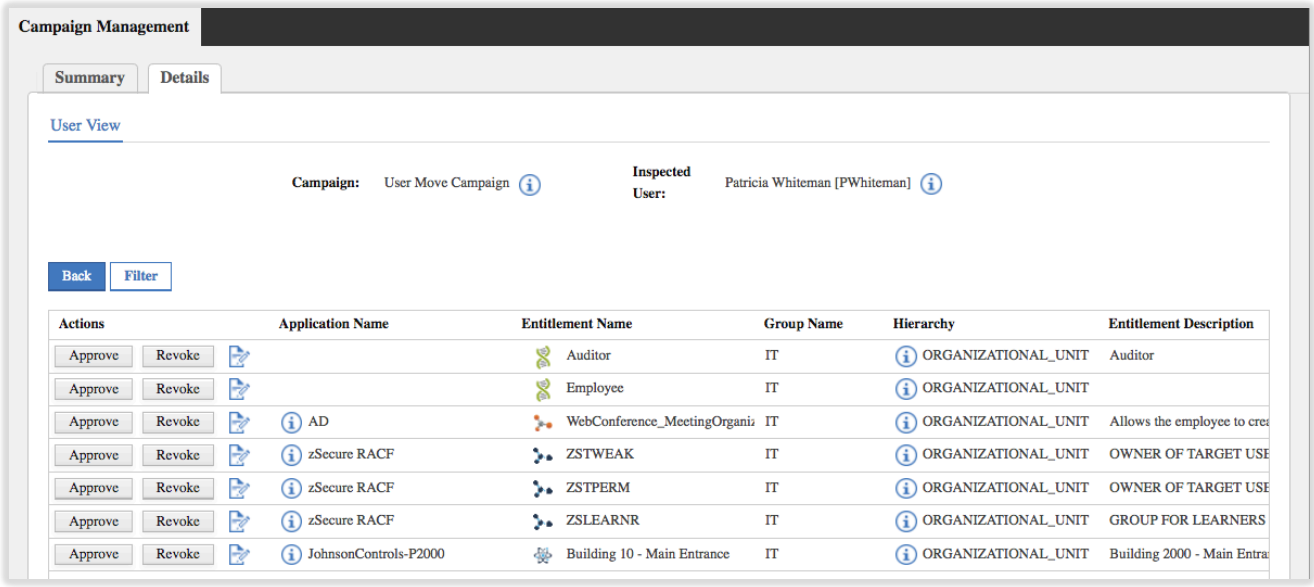
☐ Click on the new campaign (like `User Move Campaign`).

You should see Patricia showing.




☐ Click the **Inspect** icon (spyglass)

You should see all of Patricia's entitlements.




You could approve/revoke access from here to complete the process. As there are issues with the account configuration you will not see the changes sent to the target.

### 4.2.4.3 Diagnosing Issues

This section lists some of the issues you may encounter testing this scenario and how to investigate them.

Recall that the technical flow is:

- Manager logs into the Service Center and accesses the workflows available to him/her based on the workflow activities assigned to their Admin Role in the Access Requests module.
- Manager selects the new Request Move User tab, which is running the generate activity in the User Move workflow process. The form presented is based on the activity configuration which is using the user virtual view we setup.
- Manager changes the department (OU) and submits the request. As there is no approval activity (authorization) in the workflow, IGI processes the request and writes it to the IN queue as User Move and User Modify events.

- The rules engine runs and processes each event. In our case the user move rule flow will write the user and their entitlements to the campaign dataset and move the user. Once the rule flow has run successfully, the status of the events is set to Success.
- The manager logs into the Service Center and sees the user/entitlements in the Move User certification campaign. They can then approve/revoke access.

Looking at this flow, if you can select the user and request the department change in the Service Center, then the workflow and user virtual view are working. If you didn't see the menu tab or the user attribute view on your request user modify tab isn't correct, you should re-check your workflow or user virtual view configuration.

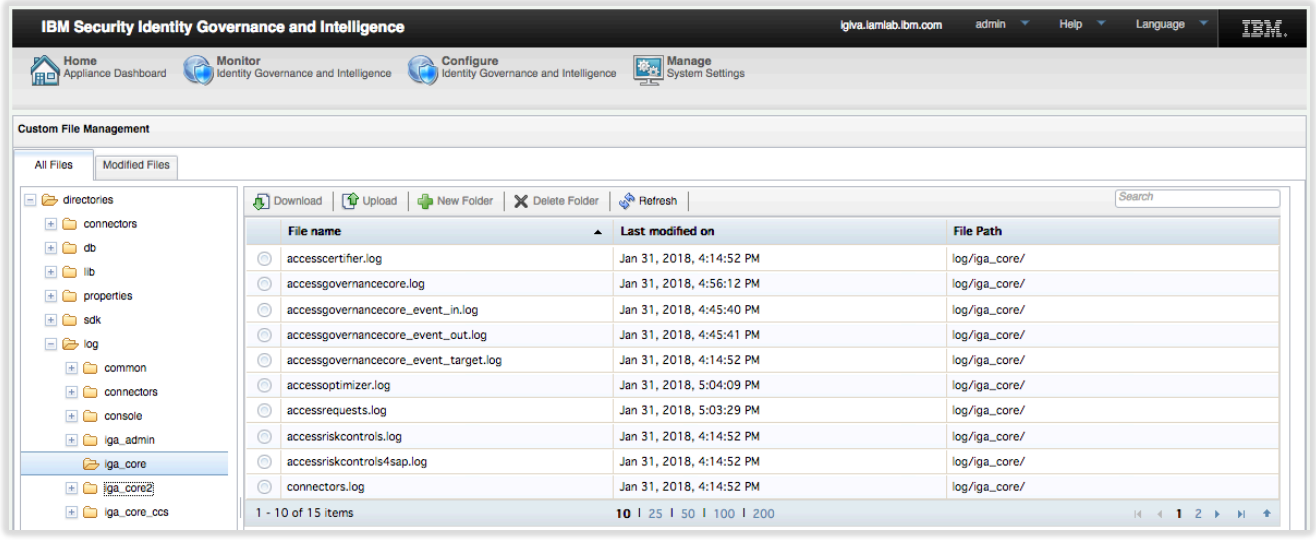
If the event appears in the `IN - User Events` monitor view, then IGI has written the event.

- If the Status is `Unprocessed`, then there may be a timing issue with the VA or the rules engine may still be down. Check for the time drift issue, and that the Rules engine is running
- If the Status is `Error`, then there may have been an unexpected or unhandled error in the rule flow.
- If the Status is `Success`, then IGI has run the rules for this event. If you're not seeing the user in the campaign, the rule flow may have had a problem that was handled.

If the event is in error or was successful, you will need to go look at the relevant IGI logs. To do so:

☐ Open the IGI VA LMI (https://192.168.42.61:9443) with admin/Passw0rd! (note the exclamation mark)
☐ Go to Configure > Custom File Management
☐ Expand the tree to log > iga_core

You should see the IGI application log files




There is a common AGC log, `accessgovernancecore.log`, and one for each of the queues, such as `accessgovernancecore_event_in.log`. You can select and download each log.

For the scenario we have here, the `accessgovernancecore_event_in.log` and `accessgovernancecore.log` are likely to give the best information.

For example, the event_in log will show details of the two events (USER_MOVE and USER_MODIFY).

```
Jan 31, 2018, 6:45:35 AM INFO  AGC:? - ************************************************************
************************************************************
Jan 31, 2018, 6:45:35 AM INFO  AGC:? - START
                     - Event propagation MR_IN_7770_2085
Jan 31, 2018, 6:45:35 AM INFO  AGC:? - Evento: [ID=7770, OPERATION=12, TRACE=null, STATE=-1,
DATEPROCESS=2018-01-31 06:45:25.0, DATEEVENT=2018-01-31 06:45:25.0, EXTTABLE=2085, ERC=user_erc,
EXTATTR1=null, EXTATTR2=IT, EXTATTR3=0, EXTATTR4=0, EXTATTR5=null, EXTATTR6=null, EXTATTR7=null,
EXTATTR8=null, EXTATTR9=null, EXTATTR10=null, OWNERSHIP=IGACORE ]
Jan 31, 2018, 6:45:35 AM INFO  AGC:? - START
Jan 31, 2018, 6:45:38 AM INFO  AGC:? - START
                     RuleFlow -> SYSTEM/IN/USER_BEFORE/RUN
Jan 31, 2018, 6:45:38 AM INFO  AGC:? - Inserito oggetto
com.engiweb.ruleengine.common.bean.ContainerBean: {}
Jan 31, 2018, 6:45:38 AM INFO  AGC:? - Inserito oggetto
com.engiweb.profilemanager.common.bean.event.EventInBean: [ID=7770, OPERATION=12, TRACE=null, STATE=-
1, DATEPROCESS=2018-01-31 06:45:25.0, DATEEVENT=2018-01-31 06:45:25.0, EXTTABLE=2085, ERC=user_erc,
EXTATTR1=null, EXTATTR2=IT, EXTATTR3=0, EXTATTR4=0, EXTATTR5=null, EXTATTR6=null, EXTATTR7=null,
EXTATTR8=null, EXTATTR9=null, EXTATTR10=null, OWNERSHIP=IGACORE ]
Jan 31, 2018, 6:45:38 AM INFO  AGC:? - Inserito oggetto
com.engiweb.profilemanager.common.bean.UserErcBean: {EMAIL=pwhiteman@igi.ibm.com, NATION=CA,
ACTION_TYPE_LAST=null, COUNTRY=null, ID=2085, SCHEDULE=0, BIRTHDAY=null, ACTION_CAUSE=0,
GIVEN_NAME=Patricia, BIRTH_COUNTRY=null, DISABLED=0, POST_EVENT=0, CURRENTOU=null,
USER_TYPE=Employee, AD_OU=null, ADDRESS=null, PHONE_NUMBER=null, ACCOUNT_EXPIRY_DATE=null,
IDENTIFICATION_NUMBER=null, LAST_MOD_TIME=2015-11-13, LAST_MOD_USER=null, DELETED=0, ATTR9=null,
ATTR4=ACME Engineering Ltd., SKIP=0, GENDER=null, ATTR3=Tertiary, BIRTH_PLACE=null, ATTR2=N,
ATTR1=DFox, ATTR8=null, PM_CODE=PWhiteman, ATTR7=I, ATTR6=null, ACTION_TYPE=0, ATTR5=null,
ATTR12=null, ATTR56=null, ATTR13=null, SURNAME=Whiteman, ATTR10=Auditor, ATTR11=null, OU=IT,
ATTR14=null, ZIPCODE=null, ATTR15=null, ACTION_CAUSE_LAST=null, CITY=San Diego, PROCESSED=0}
Jan 31, 2018, 6:45:38 AM INFO  AGC:? - STOP
                     Process IN/USER_BEFORE/RUN
Jan 31, 2018, 6:45:38 AM INFO  AGC:? - STOP
Jan 31, 2018, 6:45:38 AM INFO  AGC:? - START
Jan 31, 2018, 6:45:38 AM INFO  AGC:? - START
                     RuleFlow -> SYSTEM/IN/USER_MOVE/BEFORE
Jan 31, 2018, 6:45:38 AM INFO  AGC:? - STOP
                     Process IN/USER_MOVE/BEFORE
```

This is showing the rule flow setup and USER_BEFORE. You can see details of the eventInBean and UserErcBean.

```
Jan 31, 2018, 6:45:39 AM INFO  AGC:? - START
                      RuleFlow -> SYSTEM/IN/USER_MOVE/RUN
Jan 31, 2018, 6:45:39 AM INFO  AGC:? - Inserito oggetto
com.engiweb.ruleengine.common.bean.ContainerBean: {}
Jan 31, 2018, 6:45:39 AM INFO  AGC:? - Inserito oggetto
com.engiweb.profilemanager.common.bean.event.EventInBean: [ID=7770, OPERATION=12, TRACE=null,
STATE=1, DATEPROCESS=2018-01-31 06:45:25.0, DATEEVENT=2018-01-31 06:45:25.0, EXTTABLE=2085,
ERC=user_erc, EXTATTR1=null, EXTATTR2=IT, EXTATTR3=0, EXTATTR4=0, EXTATTR5=null, EXTATTR6=null,
EXTATTR7=null, EXTATTR8=null, EXTATTR9=null, EXTATTR10=null, OWNERSHIP=IGACORE ]
Jan 31, 2018, 6:45:39 AM INFO  AGC:? - Inserito oggetto
com.engiweb.profilemanager.common.bean.UserErcBean: {EMAIL=pwhiteman@igi.ibm.com, NATION=CA,
ACTION_TYPE_LAST=null, COUNTRY=null, ID=2085, SCHEDULE=0, BIRTHDAY=null, ACTION_CAUSE=0,
GIVEN_NAME=Patricia, BIRTH_COUNTRY=null, DISABLED=0, POST_EVENT=0, CURRENTOU=null,
USER_TYPE=Employee, AD_OU=null, ADDRESS=null, PHONE_NUMBER=null, ACCOUNT_EXPIRY_DATE=null,
IDENTIFICATION_NUMBER=null, LAST_MOD_TIME=2015-11-13, LAST_MOD_USER=null, DELETED=0, ATTR9=null,
ATTR4=ACME Engineering Ltd., SKIP=0, GENDER=null, ATTR3=Tertiary, BIRTH_PLACE=null, ATTR2=N,
ATTR1=DFox, ATTR8=null, PM_CODE=PWhiteman, ATTR7=I, ATTR6=null, ACTION_TYPE=0, ATTR5=null,
ATTR12=null, ATTR56=null, ATTR13=null, SURNAME=Whiteman, ATTR10=Auditor, ATTR11=null, OU=IT,
ATTR14=null, ZIPCODE=null, ATTR15=null, ACTION_CAUSE_LAST=null, CITY=San Diego, PROCESSED=0}
Jan 31, 2018, 6:45:39 AM INFO  AGC:? - Inserito oggetto
com.engiweb.profilemanager.common.bean.UserBean: [ID=2089, DN=null, ATTR1=null, ATTR2=null,
ATTR3=null, ATTR4=null, ATTR5=null, VALUE=null, CODE=PWhiteman, SURNAME=Whiteman, NAME=Patricia,
EMAIL=pwhiteman@igi.ibm.com, PASSWORD=null, CODFISC=null, SEX=null, DATEOFBIRTH=null,
PLACEOFBIRTH=null, ADDRESS=null, LOCALITY=null, REGISTER=null, DESCRIPTION=null, STATE=0,
LASTMODUSER=null, LASTMODTIME=Sun Feb 19 05:11:23 CET 2017, PWDMANAGEMENT_DISABLED=0,
PWDMANAGEMENT_EXPIRE=null, PWDMANAGEMENT_ID=2089, ORGANIZATIONALUNIT_ID=118,
ORGANIZATIONALUNIT_CODE=AUDIT, ORGANIZATIONALUNIT_NAME=AUDIT, HIERARCHY_ID=1, HIERARCHY_CODE=null,
HIERARCHY_NAME=null, PERSONTYPE_ID=100, PERSONTYPE_NAME=Employee, PERSONTYPE_DESCRIPTION=Identity
registered in ISIM, MASTER_CODE=null, MASTER_ID=null, UMETYPE=NONE]
Jan 31, 2018, 6:45:39 AM INFO  AGC:? - Inserito oggetto
com.engiweb.profilemanager.common.bean.ExternalInfo: [[name=OU value=IT objectValue=IT
required=null], [name=id value=2085 objectValue=2085 required=null], [name=City value=San Diego
objectValue=San Diego required=null], [name=Education - Certification value=Tertiary
objectValue=Tertiary required=null], [name=Manager value=DFox objectValue=DFox required=null],
[name=Position value=I objectValue=I required=null], [name=Is Dep. Manager value=N objectValue=N
required=null], [name=Department value=ACME Engineering Ltd. objectValue=ACME Engineering Ltd.
required=null], [name=Cod Subarea value=null objectValue=null required=null], [name=LAST_MOD_USER
value=null objectValue=null required=null], [name=LAST_MOD_TIME value=2015-11-13 objectValue=2015-11-
13 required=null], [name=ACCOUNT_EXPIRY_DATE value=null objectValue=null required=null], [name=NATION
value=CA objectValue=CA required=null], [name=Title value=Auditor objectValue=Auditor required=0],
[name=Changed value=null objectValue=null required=0]]
Jan 31, 2018, 6:45:39 AM INFO  AGC:? - Inserito oggetto
com.engiweb.profilemanager.common.bean.OrgUnitBean: [ = COPYRIGHT = null,  = serialVersionUID = -,  =
id = 112,  = name = IT,  = code = IT,  = description = IT Department,  = parent = null,  = value =
null,  = state = null,  = attr1 = null,  = attr2 = null,  = attr3 = null,  = attr4 = null,  = attr5 =
null,  = lastModUser = null,  = lastModTime = Fri Nov 13 15:14:28 CET 2015,  = enableSOD = 0,  =
organizationalunittype_description = null,  = organizationalunittype_name = null,  =
organizationalunittype_id = null,  = organizationalunittype_code = null,  = reviewState = null,  =
person_name = null,  = person_surname = null,  = person_code = null,  = person_email = null,  =
person_id = null,  = adminList = null,  = adminByDelegationList = null,  = ownerCode = null,  =
hierarchy_id = 1,  = hierarchy_name = null, ]
Jan 31, 2018, 6:45:39 AM INFO  AGC:? - STOP
                      Process IN/USER_MOVE/RUN
Jan 31, 2018, 6:45:39 AM INFO  AGC:? - START
                      RuleFlow -> SYSTEM/IN/USER_MOVE/AFTER
Jan 31, 2018, 6:45:39 AM INFO  AGC:? - STOP
                      Process IN/USER_MOVE/AFTER
Jan 31, 2018, 6:45:39 AM INFO  AGC:? - STOP
Jan 31, 2018, 6:45:40 AM INFO  AGC:? - STOP
************************************************************
************************************************************
```

This section of log is showing the rule flow for USER_MOVE. You can see the contents of the EventInBean, UserErcBean, UserBean, ExternalInfo, and OrgUnit beans. The above example shows Patricia Whiteman being moved to IT. Notice that you don't see the names of the rules being executed, just the rule flow.

This concludes this lab.

End of Document

# Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law :**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

## COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:



If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

## Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and trademark information at ibm.com/legal/copytrade.shtml.

## Statement of Good Security Practices

IT system security involves protecting systems and information through prevention, detection and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed, misappropriated or misused or can result in damage to or misuse of your systems, including for use in attacks on others. No IT system or product should be considered completely secure and no single product, service or security measure can be completely effective in preventing improper use or access. IBM systems, products and services are designed to be part of a comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products or services to be most effective. IBM DOES NOT WARRANT THAT ANY SYSTEMS, PRODUCTS OR SERVICES ARE IMMUNE FROM, OR WILL MAKE YOUR ENTERPRISE IMMUNE FROM, THE MALICIOUS OR ILLEGAL CONDUCT OF ANY PARTY.


International Business Machines Corporation
New Orchard Road Armonk, NY 10504
Produced in the United States of America 01-2016