Quick Start Guide

CA Identity Suite - Requests

Broadcom

Author: Jeremy Miller

Revision date: 7/9/2019

Access Request Overview

The purpose of this document is to describe the access request process between Identity Portal and Identity Manager. Currently, there are no use cases in the marketplace to achieve Access Request. To understand the request process we must first examine this diagram of objects and their relationship to each other.

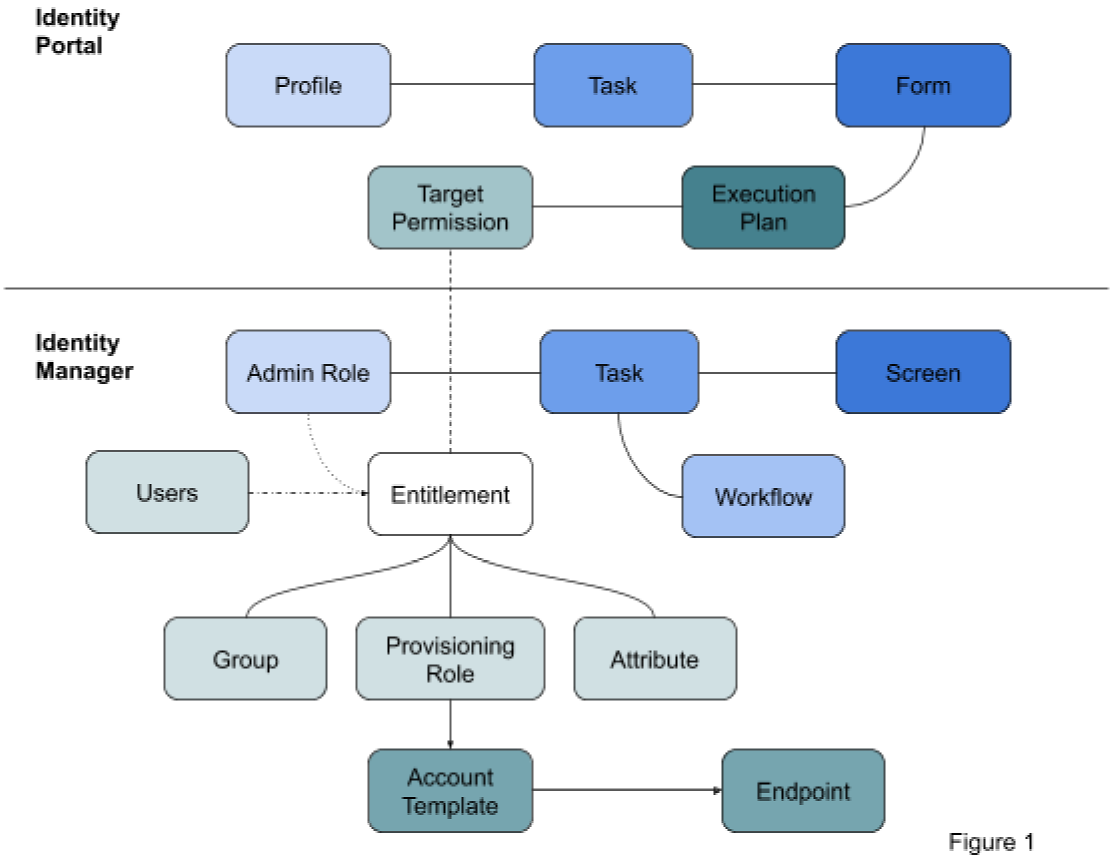


Figure 1

The basis of all requests start as a task in Identity Manager. These tasks determine key capabilities, such as if Web Services is enabled, Workflow that is attached, and what fields (attributes) are available via the screen.

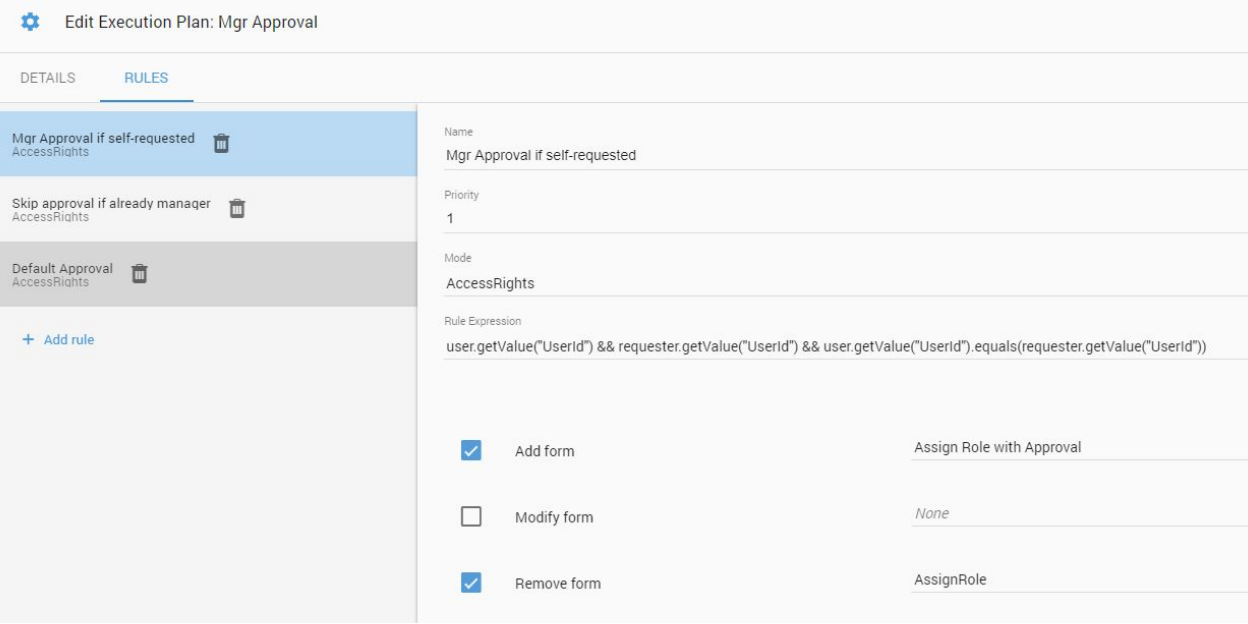
The Admin Role in Identity Manager determines Authorization. Who can access a task, and the scoping for the task when they execute. Using these two items (Admin Roles and Tasks) we can manage complex delegated administration. Example, managers can request access for all their employees, help desk can request for anyone in the organization, while anyone can do self-requests. We'll get to each of these use cases in detail later.

These Identity Manager objects Admin Roles, Tasks, Workflows, and Screens form the foundation for Identity Portal requests. In Portal, every Task maps to an Identity Task. Each Task has a Form for determining the UI layer. Currently, only attributes available in the screen are available to display in the form. There is some duplication there at the moment.

Target Permissions are a collection of entitlements. I've used "entitlements" in the diagram to represent a collection of access rights. Entitlements is a common industry term; however, they are not a physical object in Identity (only the diagram). A Target Permission (Entitlement) could be a Group, Attribute, Admin Role, or a Provisioning Role. (TODO: Access Roles, Services?)

Execution Plans might sound complex, but think of them this way. We need a way to link a Target Permission to a Form -> Task in order to provision. Since a request for a Target Permission might require varied workflows based on user type, Execution Plans allow us to implement this capability via a Rules tab.

In the example below, if the Target Permission is being self-requested, then route for Manager Approval, however, there is also a skip approval if the requester is the manager of the subject. Finally we have a default approval if the other rules do not match. Rules are an ordered list. Only one will fire.



Edit Execution Plan: Mgr Approval

Mgr Approval if self-requested
AccessRights

Skip approval if already manager
AccessRights

Default Approval
AccessRights

Name
Mgr Approval if self-requested

Priority
1

Mode
AccessRights

Rule Expression
user.getValue("UserId") && requester.getValue("UserId") && user.getValue("UserId").equals(requester.getValue("UserId"))

☑ Add form — Assign Role with Approval

☐ Modify form — *None*

☑ Remove form — AssignRole

Another example is that we might need a 2nd level of approval for privileged access rights. We can determine this based on the task / target permission and apply based on custom rules.

Access Flow

The overall flow looks like Figure 2. A Target Permission is attached to an Execution Plan. The Execution Plan determines which Form is attached (using rules) and thus which Task to call. Identity Portal makes a Web Service call to Identity Manager to the corresponding Task there. If workflow is attached it will execute, otherwise, the task will attach the entitlement. Typically the Task used here is based on a copy of Modify User task. We will walk through creation later in this document.

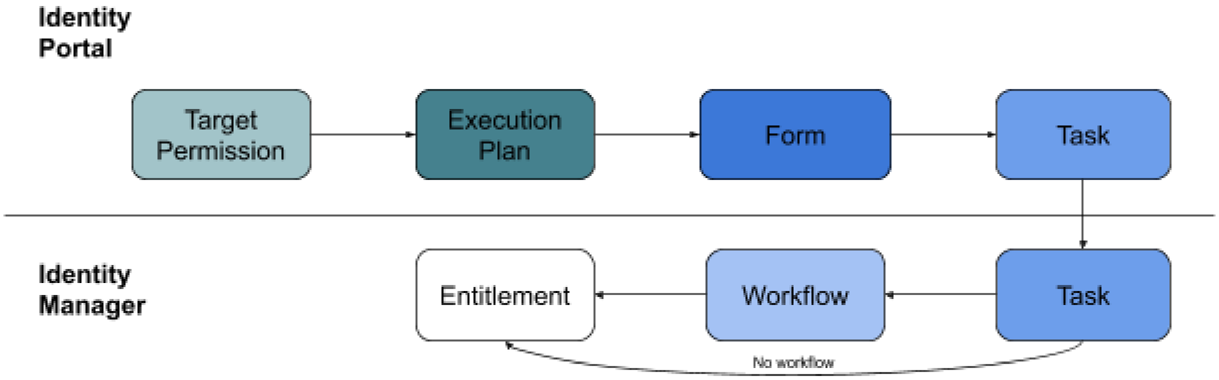


Figure 2

Access Module

The last aspect to discuss before we walk through the use cases, is the Access Module. In order to see the Access Module in portal there has to be one created. It is a built-in module, so use the quick create feature to add it if necessary. Through the Portal Admin UI select Modules at the top, if Access is not present, then Create New. Find Access and hover for quick create to show up.

This module determines a couple of authorizations. It decides who can see the module through profiles. Profile also determines Member Scope (very much like Admin Roles in Identity Manager). The module also decides which Search screens to use, filtering applied, and which attributes are returned.

A special tab for Access Rights will create our Entitlement Catalog. The catalog will only display our Target Permissions that have been configured and those we have scoping to see.

Creating an Access Request

Prerequisites: Previously deployed the Virtual Appliance with both Identity Manager and Identity Portal.

Note about scoping:
https://docops.ca.com/ca-identity-portal/14-3/EN/configuring/configuring-ca-identity-portal/scoping

Use Case - Self-request an Identity Manager group

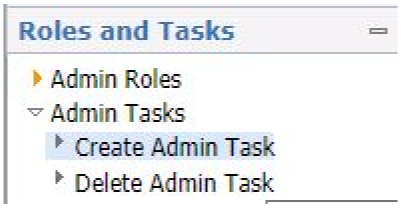
Let's get started.

We're going to start from the bottom up. We need a new task in Identity Manager.

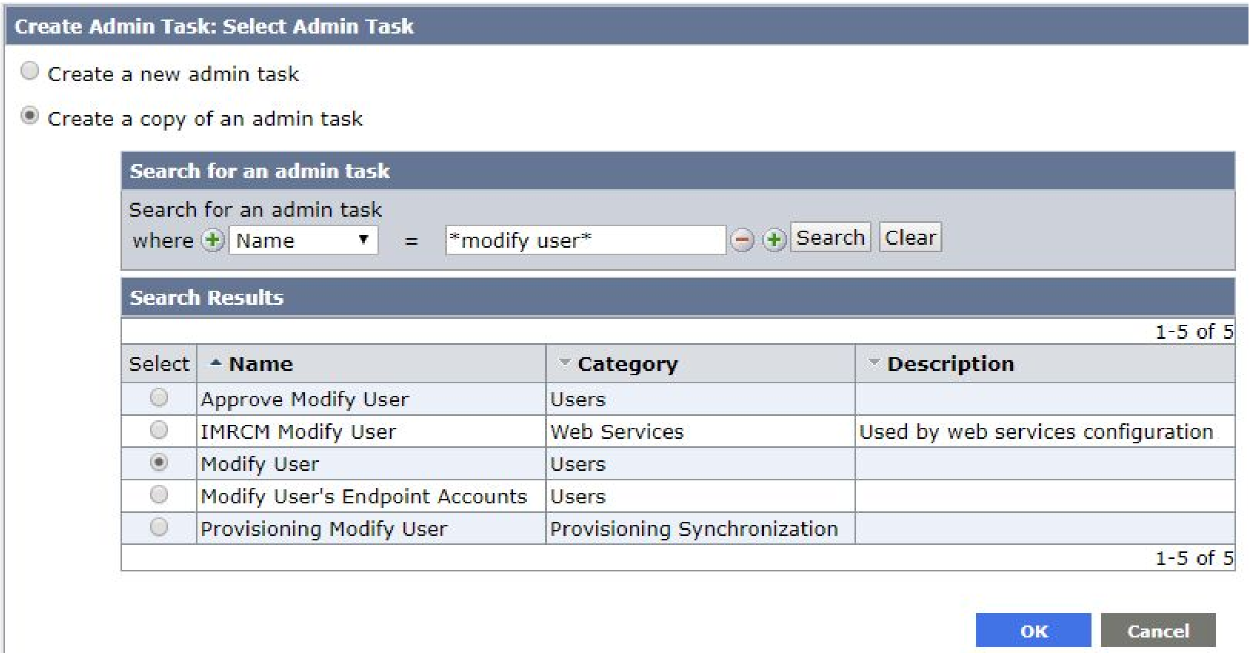
From the Identity Manager user console, login as imadmin.

Create a new Task called Self-Request

1. Roles and Tasks -> Admin Tasks -> Create Admin Task



2. Create a copy of an admin task -> Search for Modify User. Select "Modify User" then OK.



Create Admin Task: Select Admin Task

○ Create a new admin task

● Create a copy of an admin task

Search for an admin task

Search for an admin task where Name = *modify user*

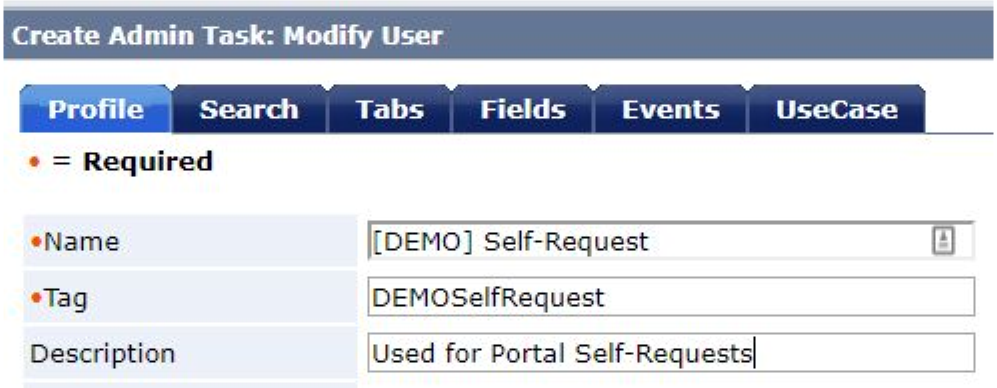
Search Results

1-5 of 5

Select	Name	Category	Description
○	Approve Modify User	Users	
○	IMRCM Modify User	Web Services	Used by web services configuration
●	Modify User	Users	
○	Modify User's Endpoint Accounts	Users	
○	Provisioning Modify User	Provisioning Synchronization	

1-5 of 5

3. Change the Name / Tag. Hint: Easier to delete the tag first, then name it. The Tag will autofill and can NOT have spaces.



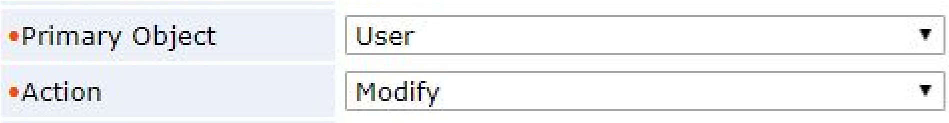
Create Admin Task: Modify User

• = Required

•Name	[DEMO] Self-Request
•Tag	DEMOSelfRequest
Description	Used for Portal Self-Requests

4. Category and Category Order are only relevant for display purposes in Identity Manager.

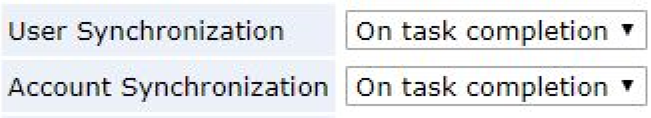
5. Primary Object and Action must be User and Modify respectively.



•Primary Object	User
•Action	Modify

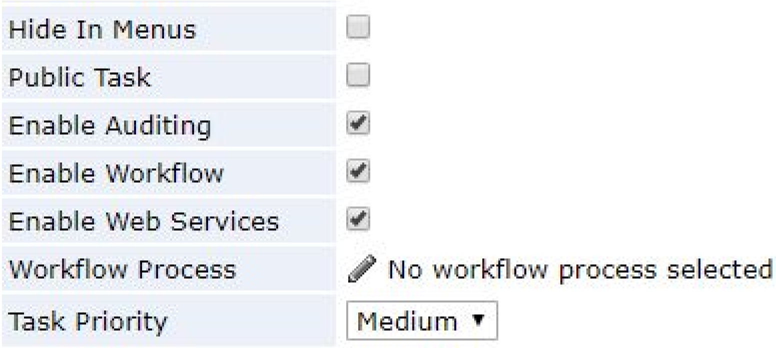
6. User and Account Synchronization must be set, or the user will not get updated properly.
   a. User Synchronization will trigger Identity Policies

b. Account Synchronization will trigger provisioning actions



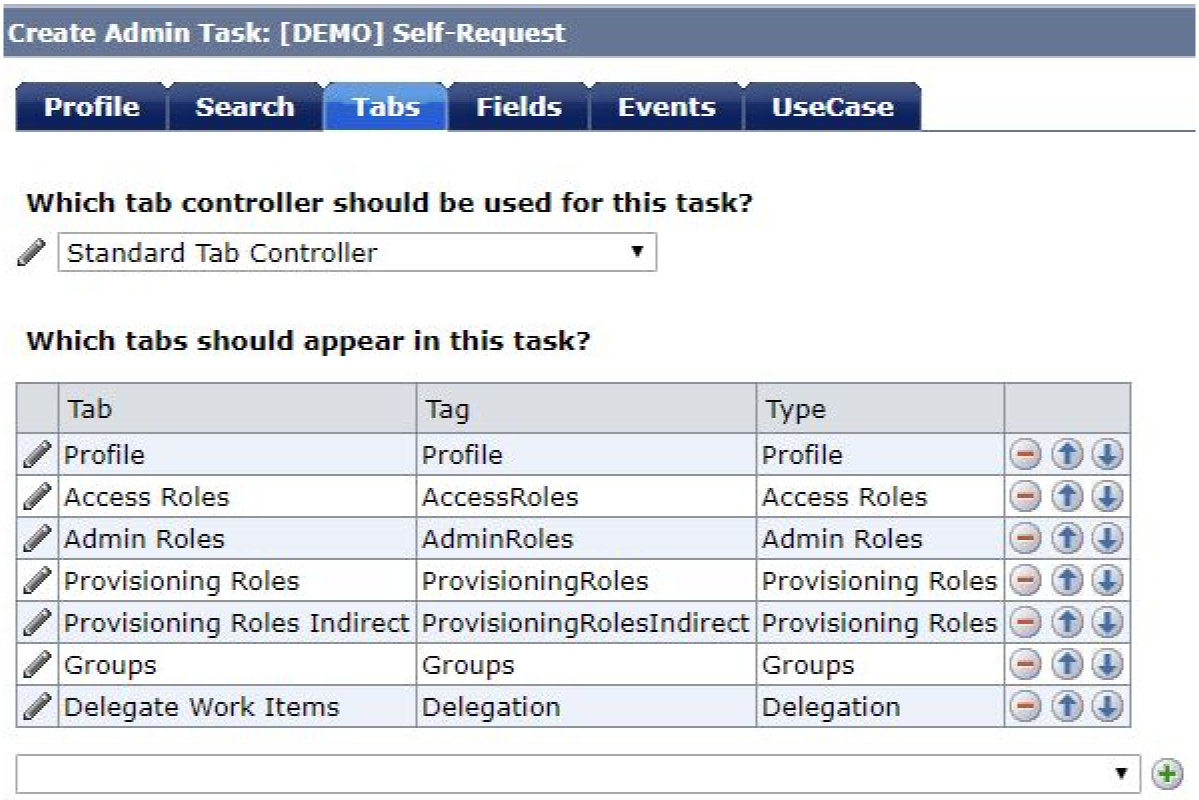
User Synchronization	On task completion ▼
Account Synchronization	On task completion ▼

7. Enable Web Services must be checked, or Portal will not be able to call the task.



Hide In Menus	☐
Public Task	☐
Enable Auditing	☑
Enable Workflow	☑
Enable Web Services	☑
Workflow Process	✎ No workflow process selected
Task Priority	Medium ▼

8. Click on the Tabs tab.



Create Admin Task: [DEMO] Self-Request

Which tab controller should be used for this task?

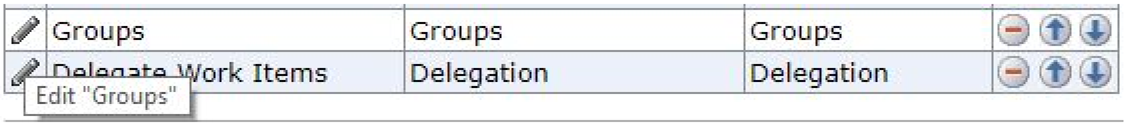
✎ Standard Tab Controller ▼

Which tabs should appear in this task?

	Tab	Tag	Type	
✎	Profile	Profile	Profile	⊖ ↑ ↓
✎	Access Roles	AccessRoles	Access Roles	⊖ ↑ ↓
✎	Admin Roles	AdminRoles	Admin Roles	⊖ ↑ ↓
✎	Provisioning Roles	ProvisioningRoles	Provisioning Roles	⊖ ↑ ↓
✎	Provisioning Roles Indirect	ProvisioningRolesIndirect	Provisioning Roles	⊖ ↑ ↓
✎	Groups	Groups	Groups	⊖ ↑ ↓
✎	Delegate Work Items	Delegation	Delegation	⊖ ↑ ↓

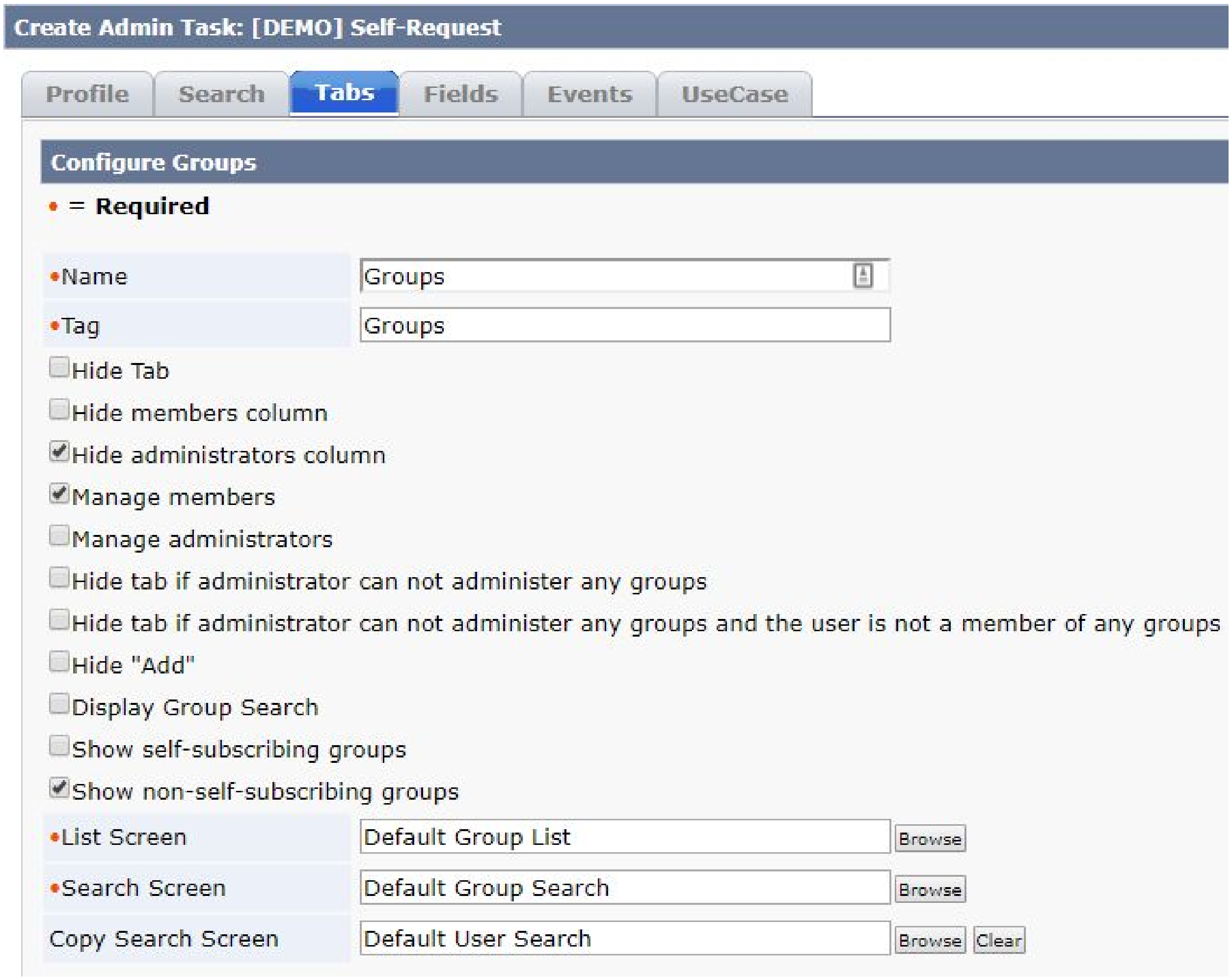
▼ ⊕

9. The profile will determine which attributes are available on the screens, but we are keeping it simple. Let’s click on **Groups**.



Groups	Groups	Groups	
Delegate Work Items	Delegation	Delegation	

10. Uncheck “Manager Administrators” and Check “Hide Administrators column”



Create Admin Task: [DEMO] Self-Request

Configure Groups

• = Required

•Name: Groups

•Tag: Groups

☐ Hide Tab

☐ Hide members column

☑ Hide administrators column

☑ Manage members

☐ Manage administrators

☐ Hide tab if administrator can not administer any groups

☐ Hide tab if administrator can not administer any groups and the user is not a member of any groups

☐ Hide "Add"

☐ Display Group Search

☐ Show self-subscribing groups

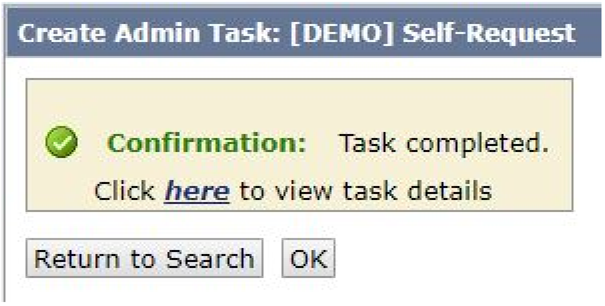
☑ Show non-self-subscribing groups

•List Screen: Default Group List [Browse]

•Search Screen: Default Group Search [Browse]

Copy Search Screen: Default User Search [Browse] [Clear]

11. Click OK.
12. Do the same procedure to the “**Provisioning Roles**” tab
13. Uncheck “Manager Administrators” and Check “Hide Administrators column”
14. Click Submit.
15. The new task should now be created.



Create Admin Task: [DEMO] Self-Request

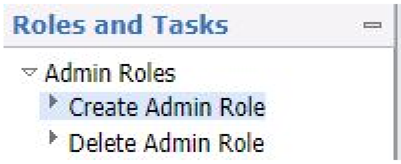
Confirmation: Task completed.

Click *here* to view task details

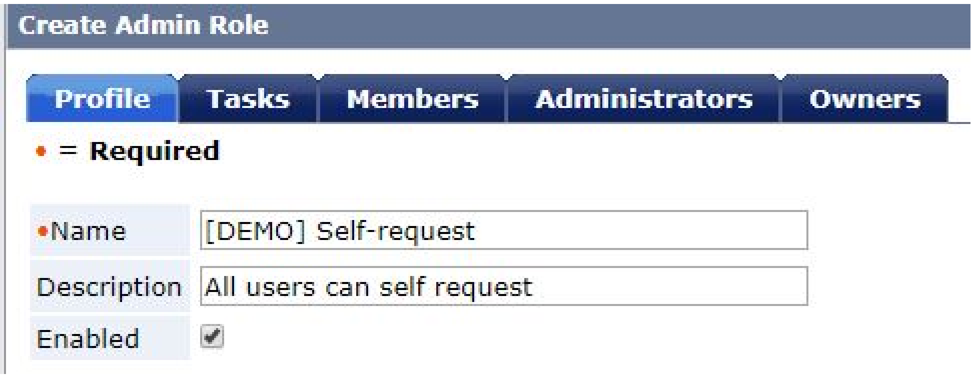
Create a new Admin Role called Self-Request

As noted above, we need to create an Admin Role. This is how scoping is achieved. The scoping allows users to see the entitlements that will be associated to the Self-Request task through the Execution Plan.

1. Roles and Tasks -> Admin Roles -> Create Admin Role



2. Create a Name, Description, and make sure to check Enabled.

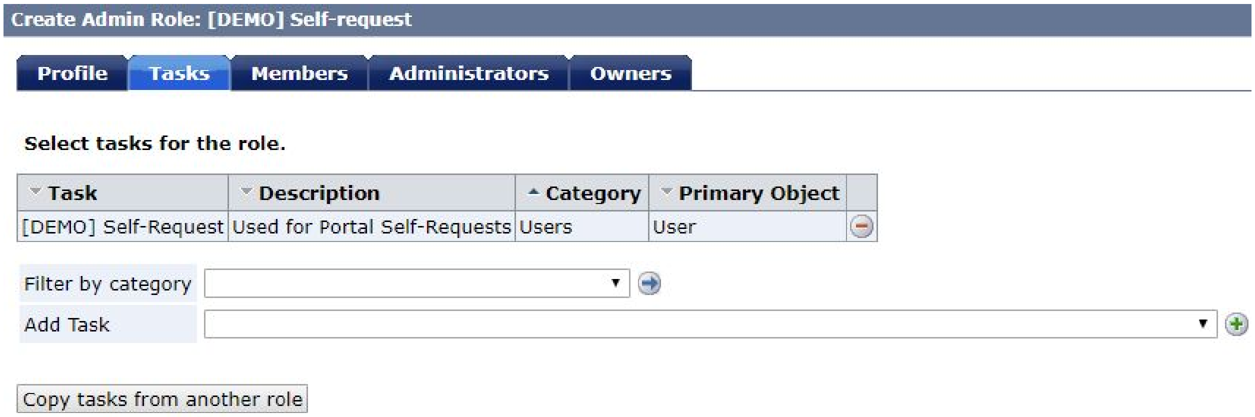


Create Admin Role

• = Required

•Name	[DEMO] Self-request
Description	All users can self request
Enabled	☑

3. Click on the Tasks tab, then add task [DEMO] Self-Request



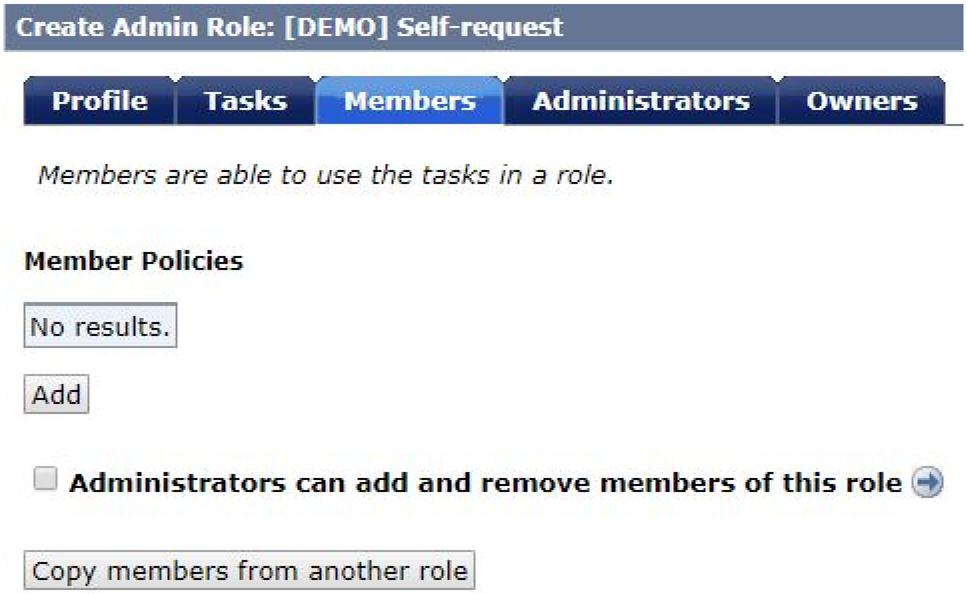
Create Admin Role: [DEMO] Self-request

Select tasks for the role.

Task	Description	Category	Primary Object	
[DEMO] Self-Request	Used for Portal Self-Requests	Users	User	⊖

Filter by category	
Add Task	

4. Click Members tab, this is where we will create our member rule.



Create Admin Role: [DEMO] Self-request

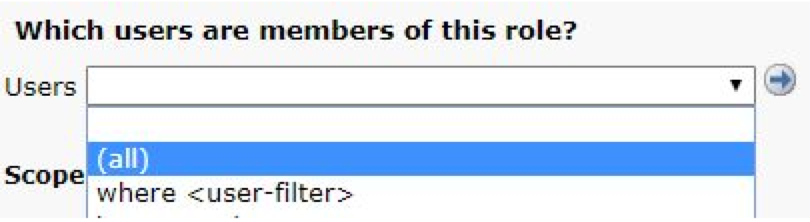
Members are able to use the tasks in a role.

Member Policies

No results.

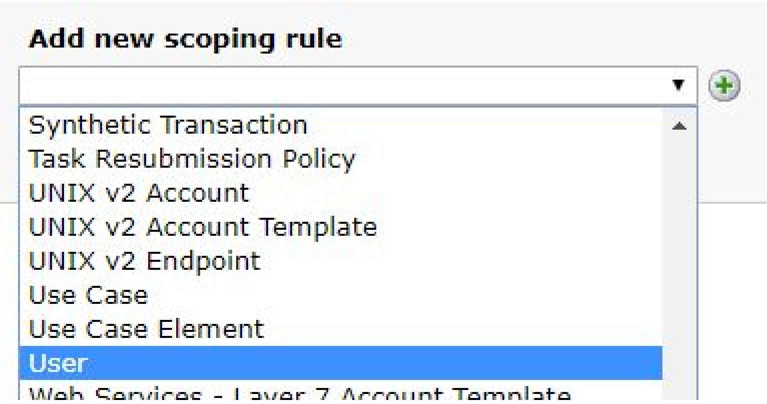
□ Administrators can add and remove members of this role

5. Click Add, in Users click (all)



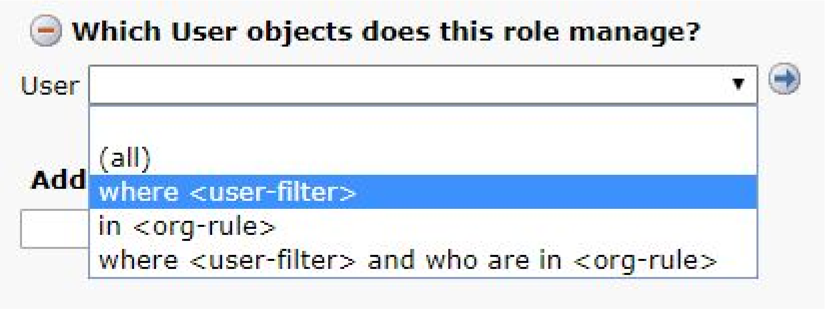
Which users are members of this role?

6. Scope Rules - Click the dropdown, click User

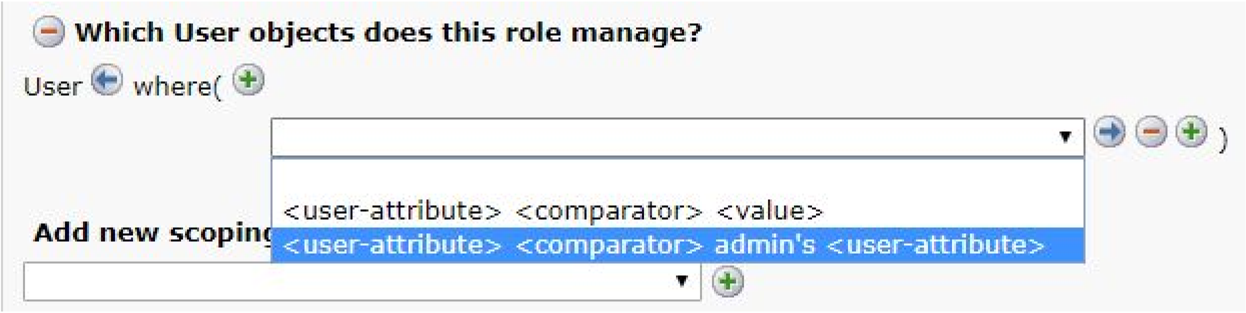


Add new scoping rule

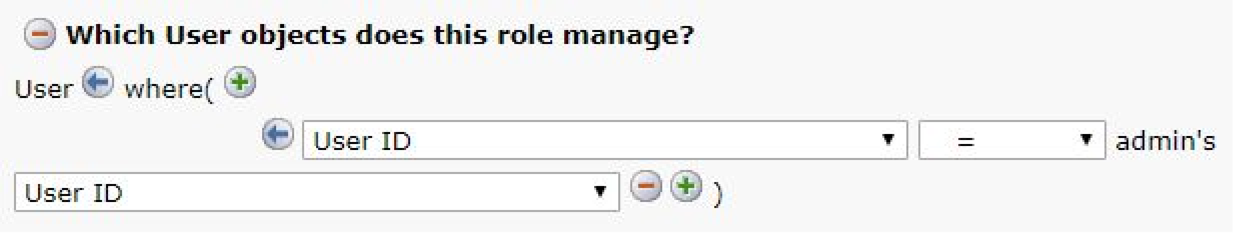
7. Which User objects does this role manage? Find and click “where <user-filter>”


Which User objects does this role manage?
User
Add
(all)
where <user-filter>
in <org-rule>
where <user-filter> and who are in <org-rule>

8. See if changes slightly, now select “<user-attribute> <comparator> admin’s <user-attribute>

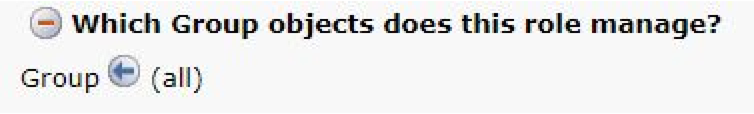

Which User objects does this role manage?
User where(
)
<user-attribute> <comparator> <value>
<user-attribute> <comparator> admin's <user-attribute>
Add new scoping

9. Select User ID for both fields. By doing so, we’re saying where the User ID of the subject equals the User ID of the Admin (meaning Self-Administration)


Which User objects does this role manage?

User where(User ID = admin's User ID)

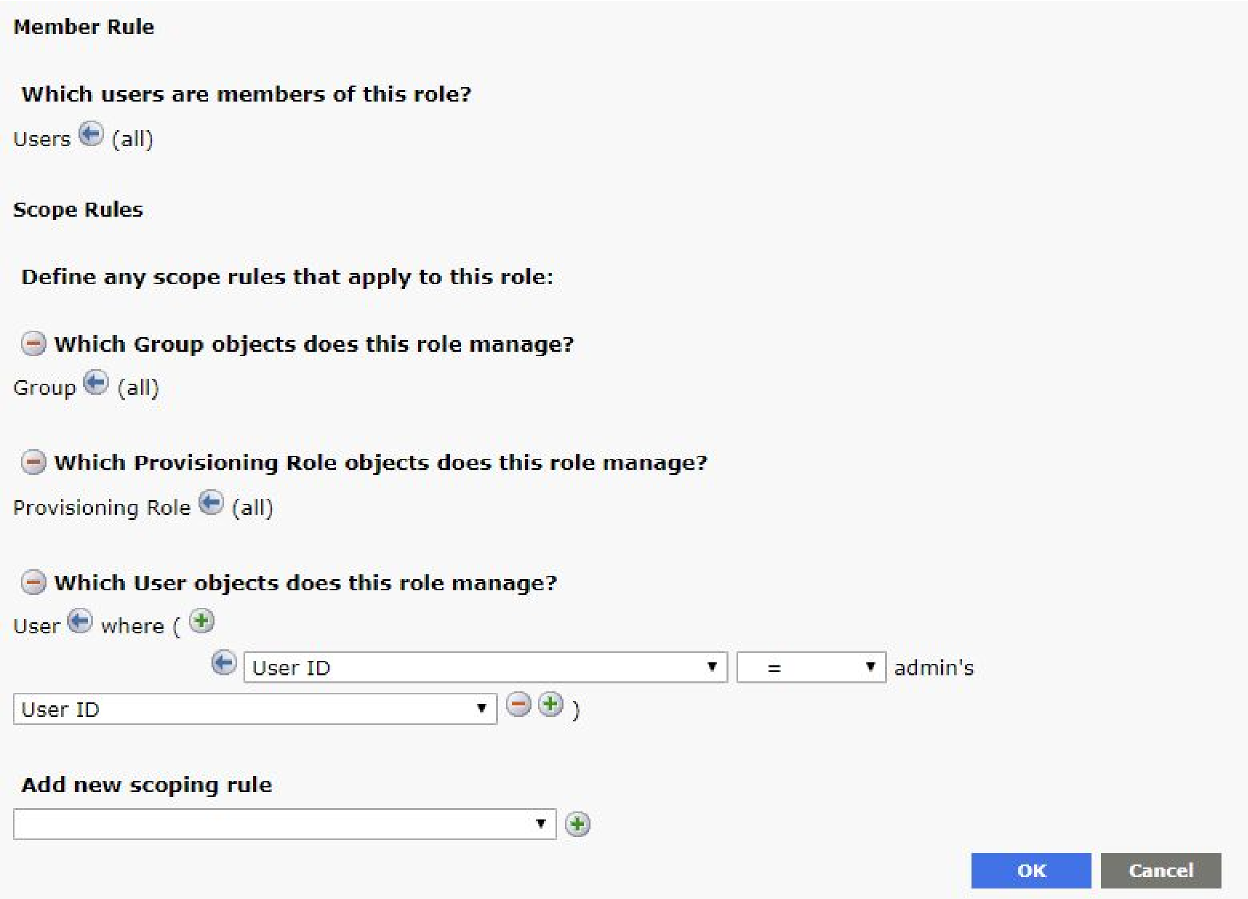
10. Then under Add new scoping rule select Group
11. Group (all)


Which Group objects does this role manage?

Group (all)

12. Repeat for Provisioning Role
13. Provisioning Role (all)

14. Member rule should look like this, then click OK



Member Rule

Which users are members of this role?

Users (all)

Scope Rules

Define any scope rules that apply to this role:

Which Group objects does this role manage?

Group (all)

Which Provisioning Role objects does this role manage?

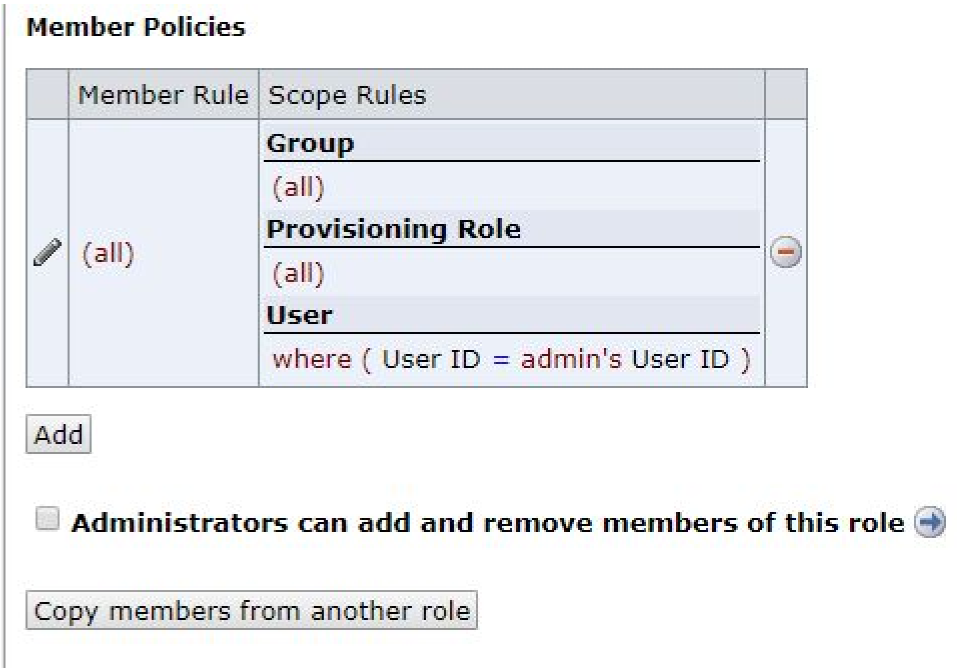
Provisioning Role (all)

Which User objects does this role manage?

User where (User ID ▾ = ▾ admin's User ID ▾)

Add new scoping rule

15. Then the Member policy will look like this.

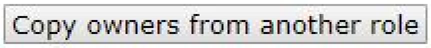


Member Policies

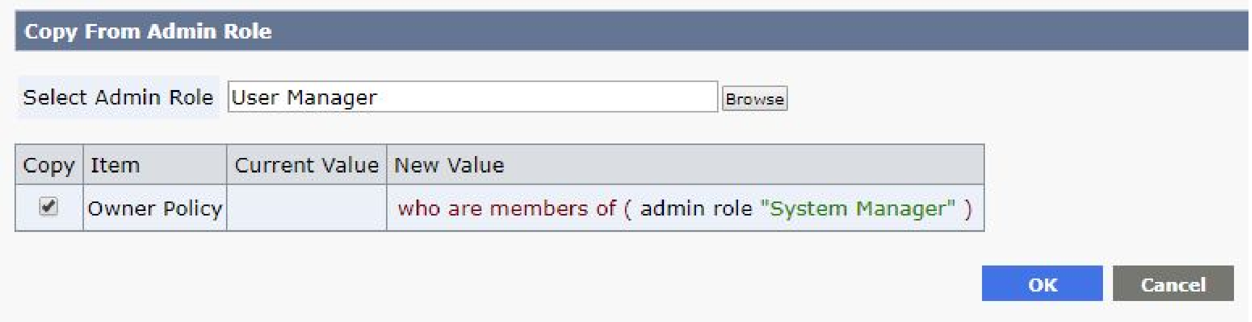
	Member Rule	Scope Rules	
	(all)	**Group** (all) **Provisioning Role** (all) **User** where (User ID = admin's User ID)	

□ **Administrators can add and remove members of this role**

16. Lastly we need an owner assigned to this Admin Role, click the Owners tab.
17. Copy from another role



18. Select User Manager, check the box next to Owner Policy

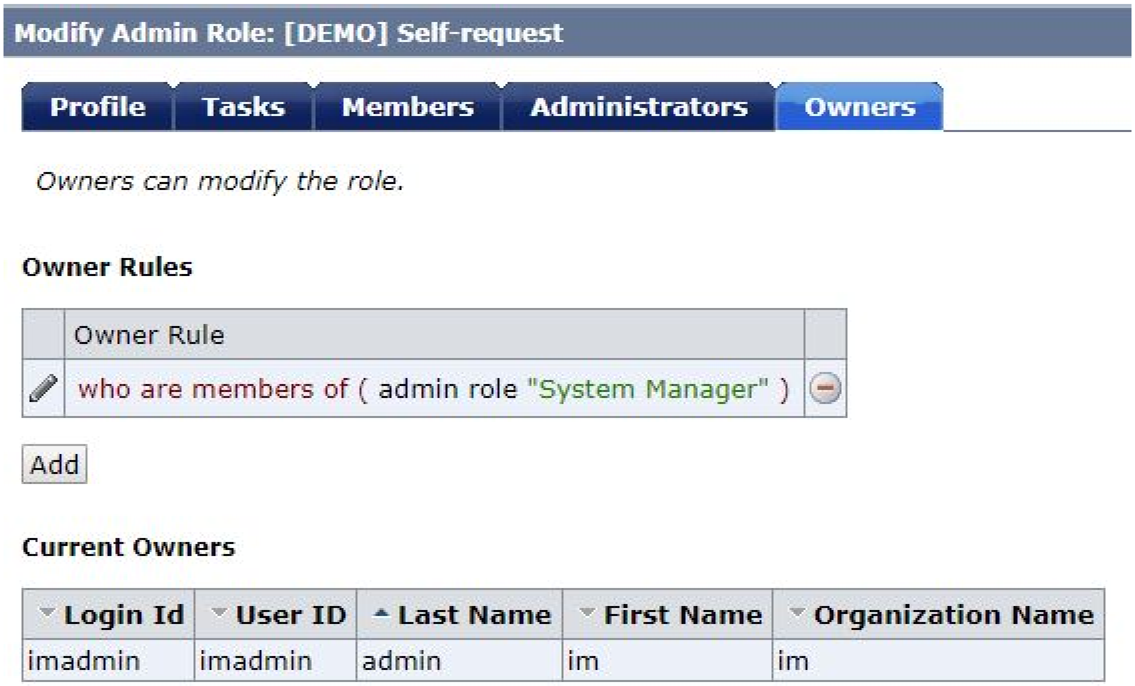


Copy From Admin Role

Select Admin Role: User Manager

Copy	Item	Current Value	New Value
☑	Owner Policy		who are members of (admin role "System Manager")

19. Final Owners tab should look like this.



Modify Admin Role: [DEMO] Self-request

Owners can modify the role.

Owner Rules

	Owner Rule	
✎	who are members of (admin role "System Manager")	⊖

Current Owners

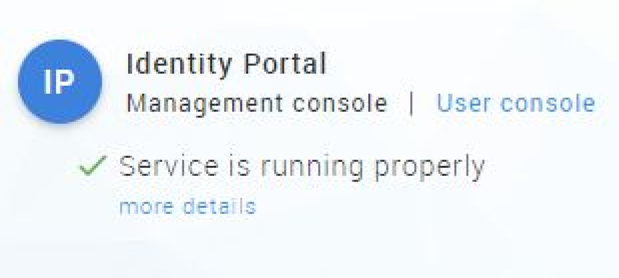
Login Id	User ID	Last Name	First Name	Organization Name
imadmin	imadmin	admin	im	im

20. Now click Submit.

What we've done is given ALL users the ability to manage themselves using the [DEMO] Self-Request task with scoping of all groups and provisioning roles.

Link the Portal Task to our Identity Manager Task

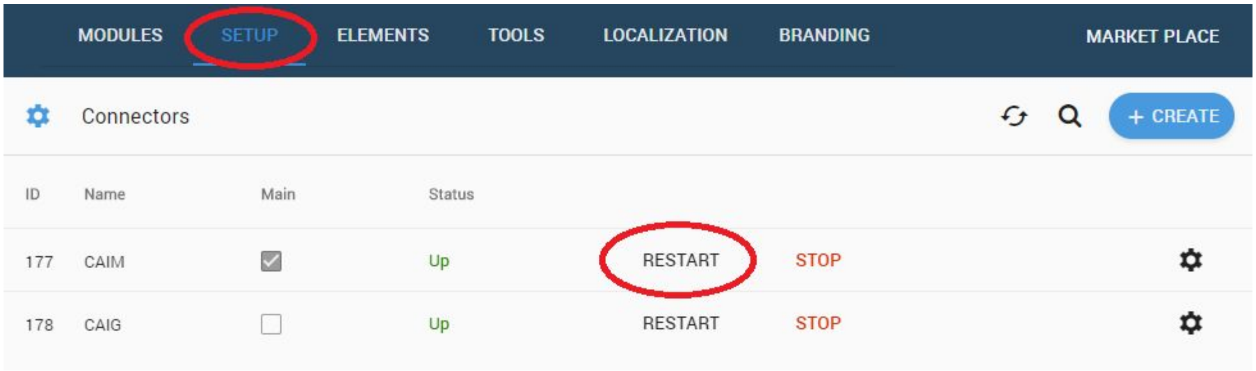
1. First we need to navigate to the Identity Portal Admin UI


IP
Identity Portal
Management console | User console
✓ Service is running properly
more details

We need to refresh the CAIM connector to pull in the latest Web Services WSDL

NOTE: Restart may not be needed in 14.2 or higher

2. Click Setup, then on the CAIM connector click Restart



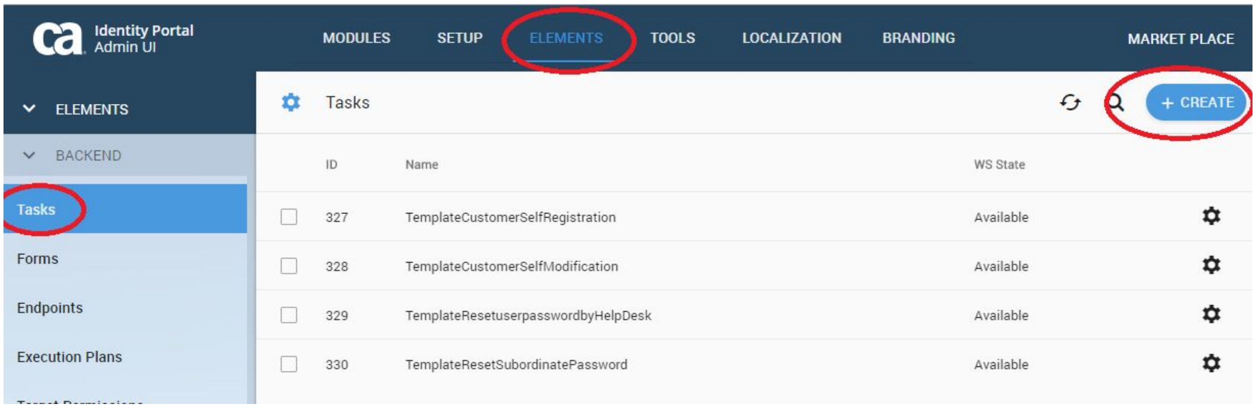
ID	Name	Main	Status			
177	CAIM	☑	Up	RESTART	STOP	⚙
178	CAIG	□	Up	RESTART	STOP	⚙

Now we'll create the task

3. Click on Elements on the top nav bar

4. Click on Tasks on the left hand nav

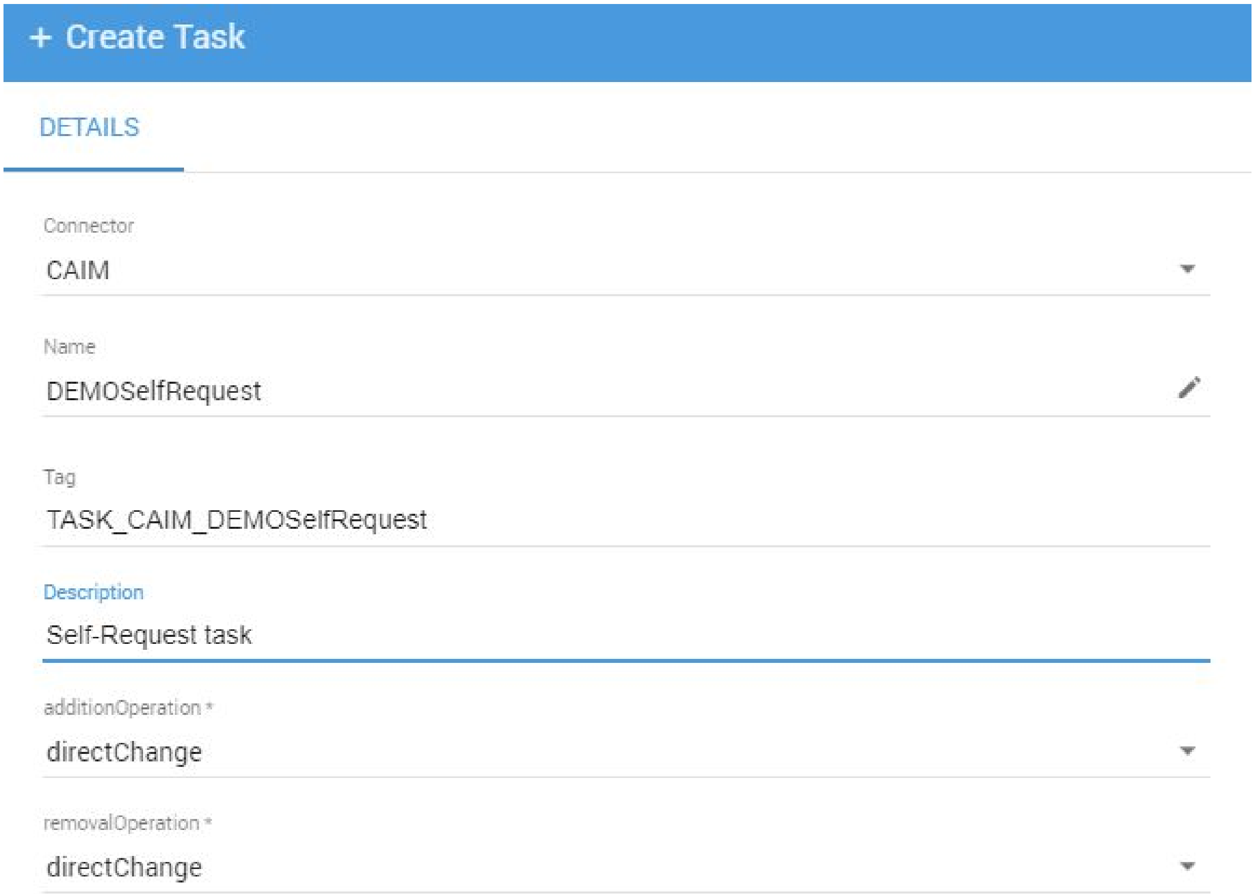
5. Click + Create



Tasks

	ID	Name	WS State
□	327	TemplateCustomerSelfRegistration	Available
□	328	TemplateCustomerSelfModification	Available
□	329	TemplateResetuserpasswordbyHelpDesk	Available
□	330	TemplateResetSubordinatePassword	Available

6. Enter the following details
    a. Connector: CAIM
    b. Name: DEMOSelfRequest (If this doesn't auto-populate as you type, backtrack and look for a missing step. It means that task isn't part of the WSDL yet)
    c. Description: Self-Request task



Connector
CAIM

Name
DEMOSelfRequest

Tag
TASK_CAIM_DEMOSelfRequest

Description
Self-Request task

additionOperation *
directChange

removalOperation *
directChange

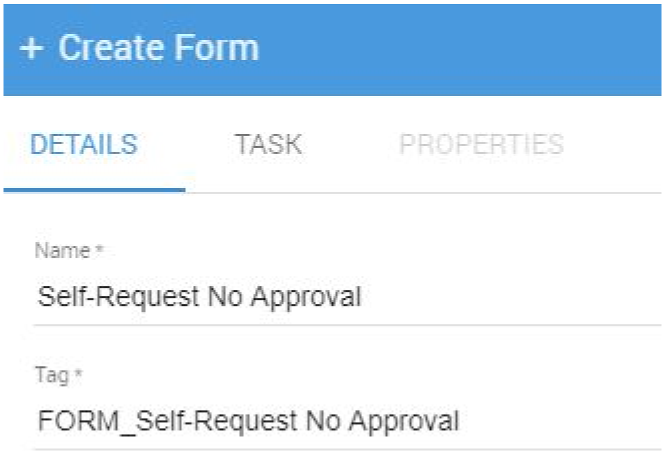
7. Click Create, then Finish

Add a Form to our Portal Task

Forms are used to link from an Execution Plan to a Task/Workflow, so we need to use a descriptive name here. If the task you are linking is an approval task, describe it here. For this use case we chose no approval.

1. In the Elements tab -> Forms, click + Create

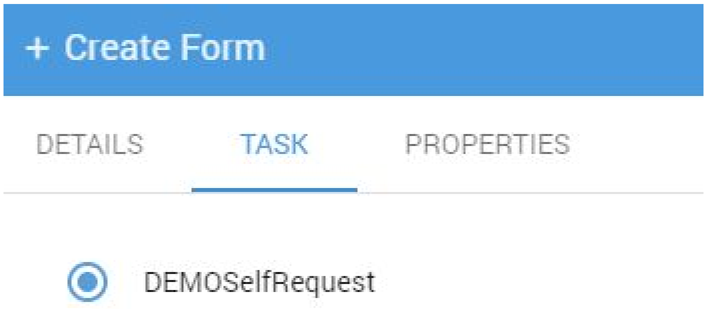
2. Name: Self-Request No Approval



Name *
Self-Request No Approval

Tag *
FORM_Self-Request No Approval

3. Click on Task and select the DEMOSelfRequest we created in the previous section.



● DEMOSelfRequest

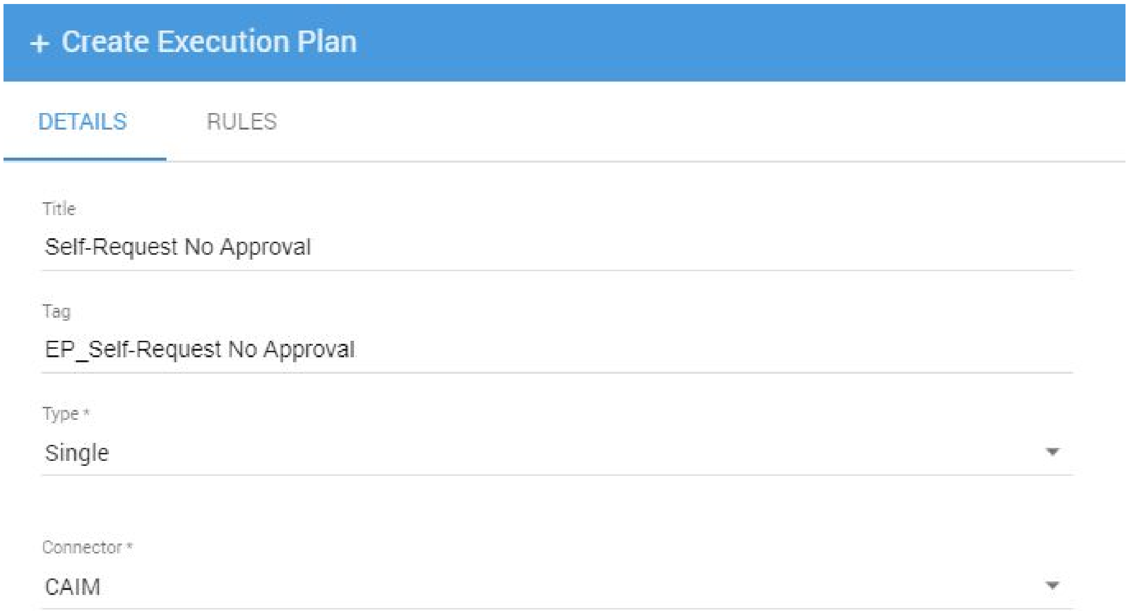
The Properties tab is where we decided what will be displayed for this form. Since we are only assigning Groups we don't need anything here. We'll cover options in other use cases below.

4. Click Create, then Finish

Create an Execution Plan and link to a Form

The Execution Plan links a Target Permission to a Form and thus a Task. This is precisely why we were descriptive in our Form creation in the last step. As we add capabilities over time, we'll want to know what Form/Task is being used and what's available for these Target Permission requests.

1. In the Elements tab -> Execution Plans, click + Create
    a. Notice the Tag gets a EP_ to designate the ExecutionPlan



+ Create Execution Plan

DETAILS RULES

Title
Self-Request No Approval

Tag
EP_Self-Request No Approval

Type *
Single

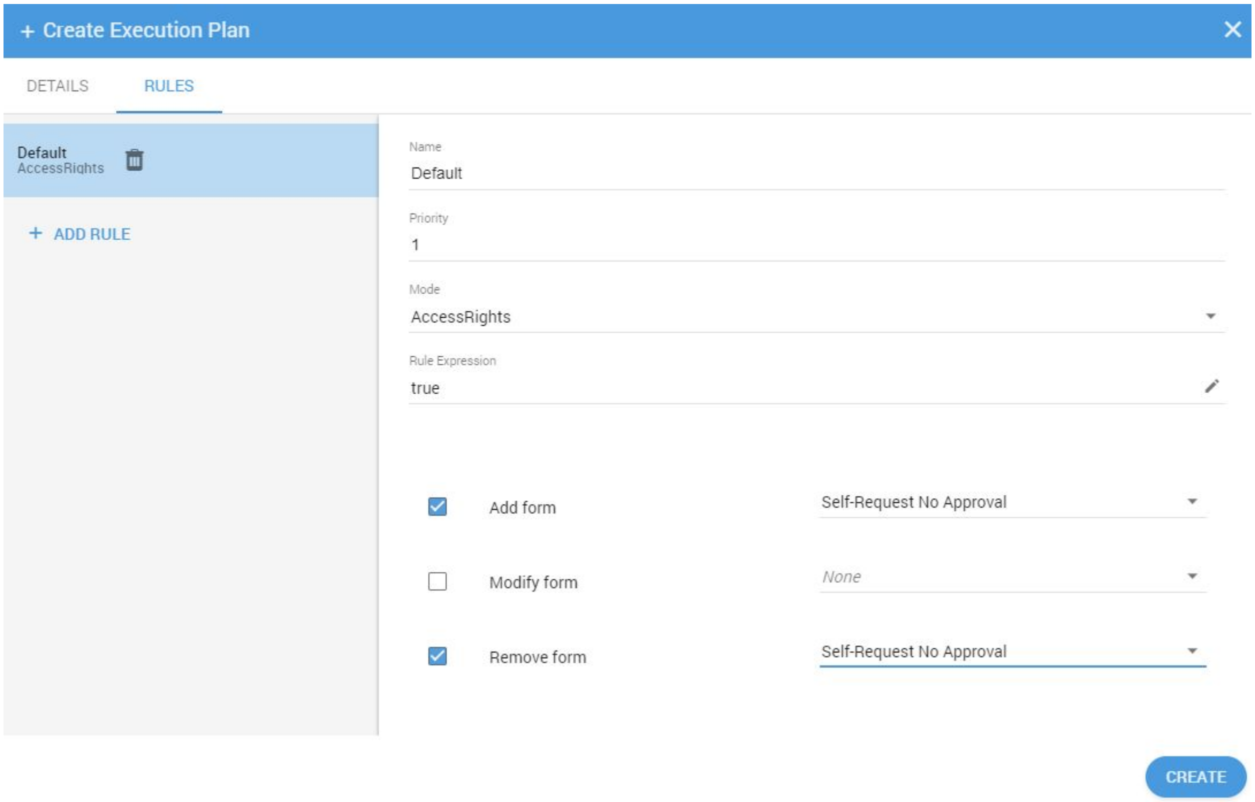
Connector *
CAIM

2. On the Rules tab click + Add Rule
    a. Here we could apply conditional policies, but in this case we'll create a default

3. Name: Default
4. Click Add Form and select our previously created form Self-Request No Approval

5. Click Remove Form and select Self-Request No Approval

6. Click Create, then Finish



Default
AccessRights

Name: Default

Priority: 1

Mode: AccessRights

Rule Expression: true

☑ Add form: Self-Request No Approval

□ Modify form: None

☑ Remove form: Self-Request No Approval

7. That completes the Execution Plan

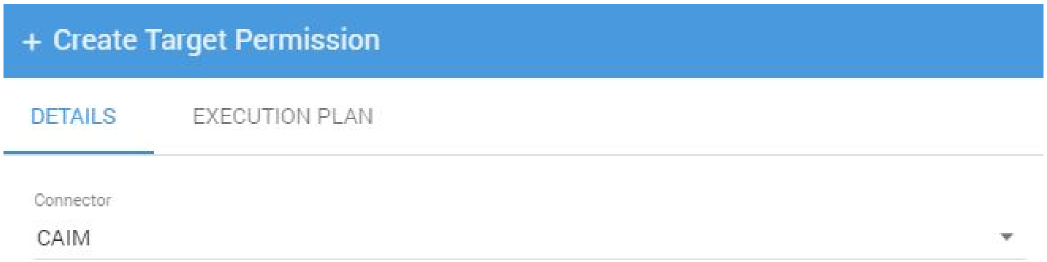
Notice that we would have nothing to select if we had not created a Form first. The form ties this Execution Plan to the Task of Demo Self-Request. There's no workflow attached to that task, thus we used the No Approval descriptive text for clarification.

Create a Target Permission

As mentioned in the pretext a target permission could be linked to a variety of Identity Manager resources. Think of this as an Entitlement, whether it be a group, role, or attribute value. An entitlement grants access to something. In our example we're going to use a provisioning role called Demo App user

1. In the Elements tab -> Target Permissions, click + Create

2. Connector: CAIM

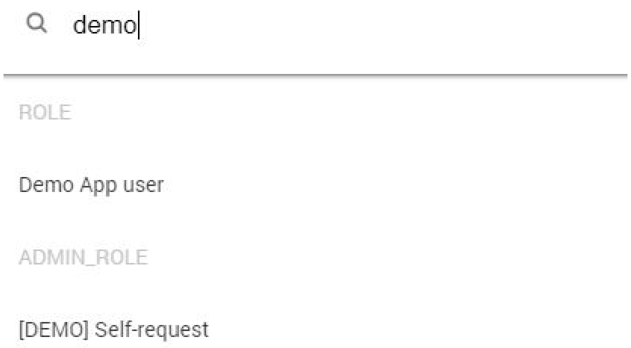


+ Create Target Permission

DETAILS EXECUTION PLAN

Connector
CAIM

3. Select Target Permission Name: Search for Demo (select Demo App user



demo

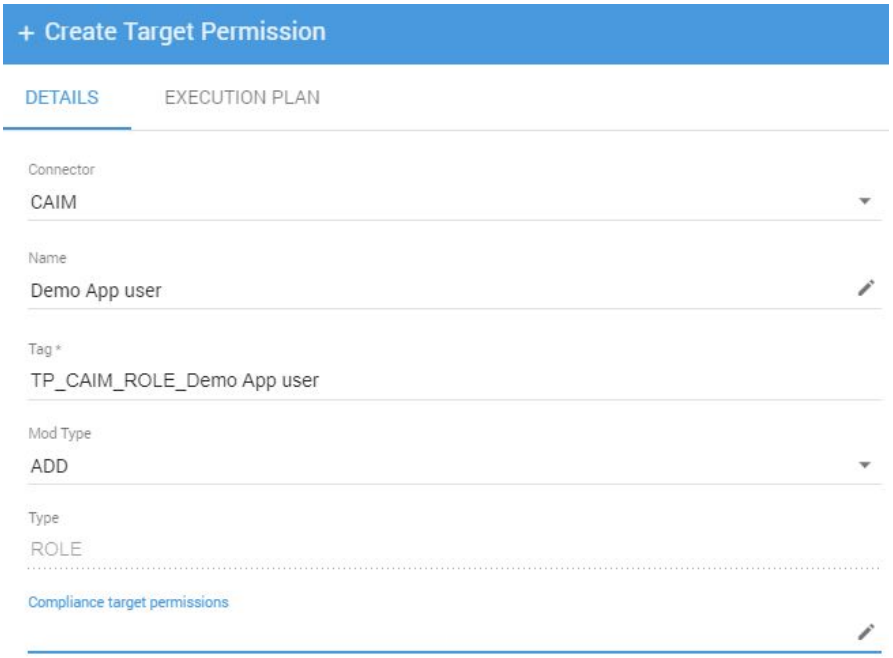
ROLE

Demo App user

ADMIN_ROLE

[DEMO] Self-request

4. Select Demo App user
   a. This will populate the remaining details



Connector: CAIM

Name: Demo App user

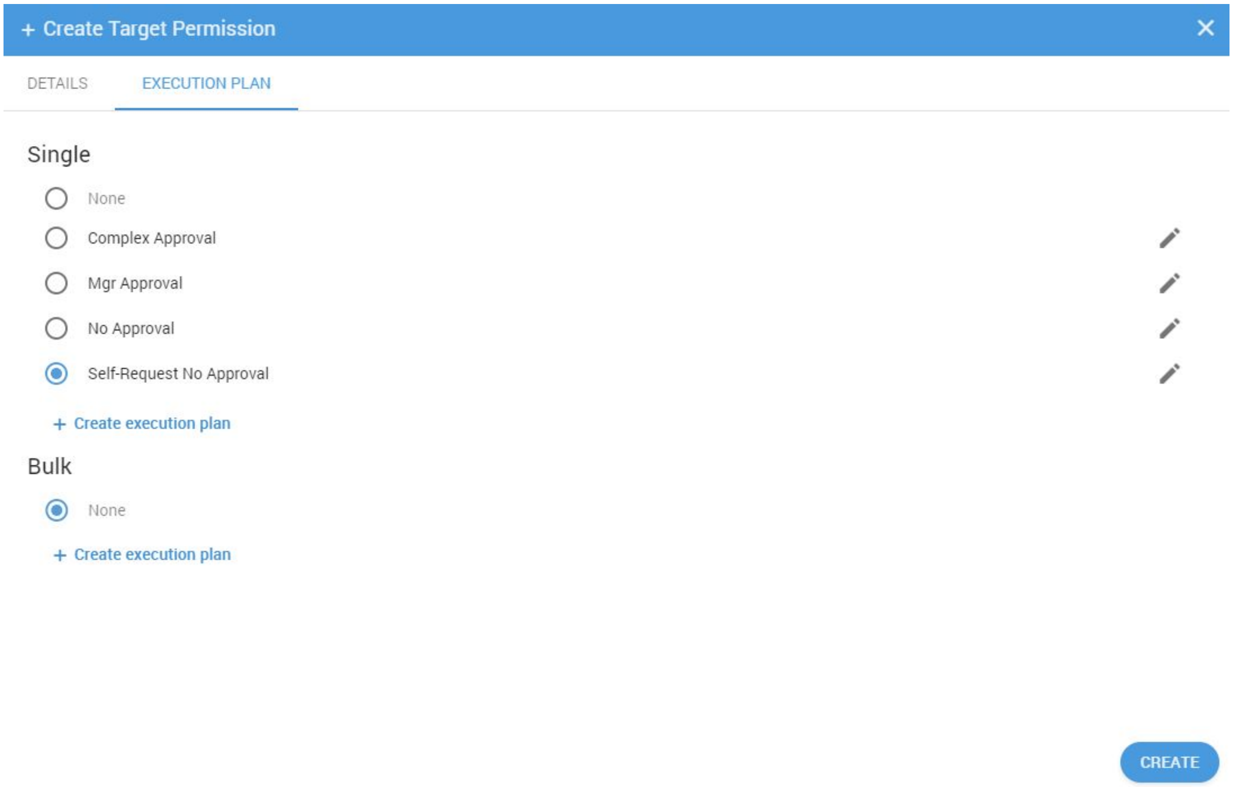
Tag *: TP_CAIM_ROLE_Demo App user

Mod Type: ADD

Type: ROLE

Compliance target permissions

5. Click on the Execution Plan tab and select our Self-Request No Approval execution plan



Single

○ None
○ Complex Approval
○ Mgr Approval
○ No Approval
● Self-Request No Approval

Bulk

● None

6. Click Create, then Finish

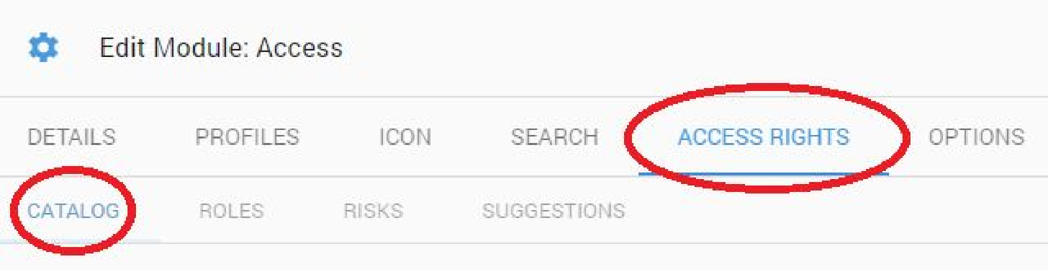
Add the Target Permission to the access catalog

With our Target Permission created and linked to an Execution Plan, we can now make it available in the access catalog for a user to request. This is done through the access module.

1. In the Modules tab, select Access

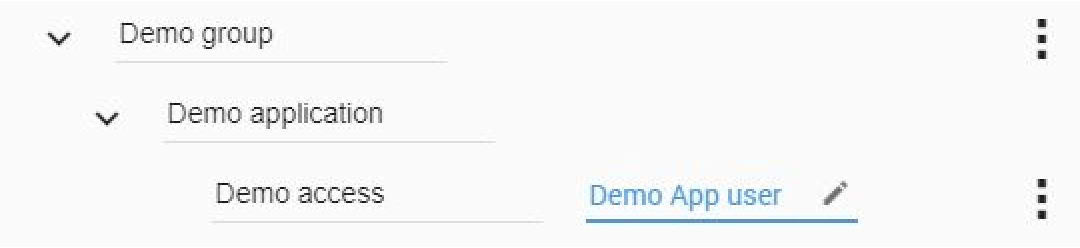
2. Select Access Rights
    a. Here you will find Catalog, Roles, Risks, and Suggestions
    b. Each of these are Portal specific capabilities

3. Select Catalog



4. Add an Application Group, Application, and finally add your permission

   Note: This is where you enter a business friendly name for the Target Permission. Make sure it makes sense to the users, because this name is all they will see in Identity Portal.



Demo group

Demo application

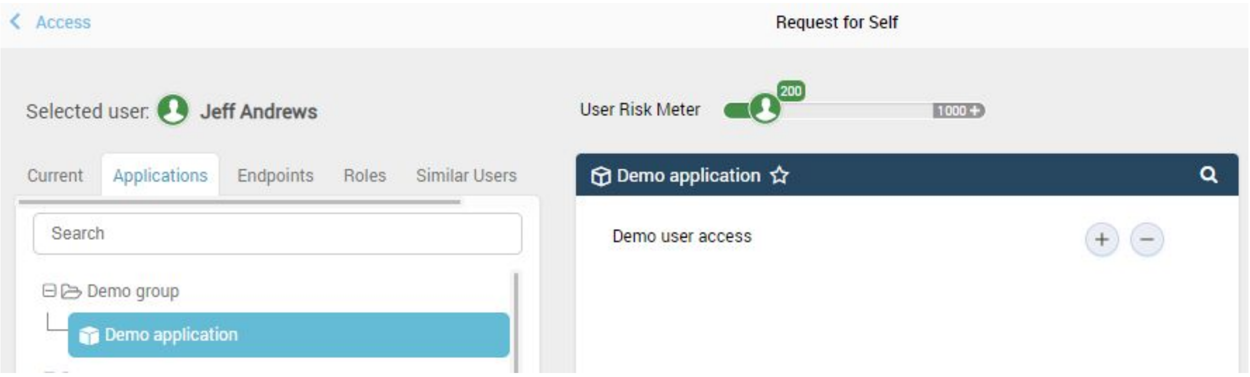
Demo access   Demo App user

5. Click Save in the upper right

Resulting Access Request

The finished Product should look like this in the Access Request

1. Login as an end user
2. Select Access
3. Click Applications (so you’re not looking at current access only)
4. Search for demo application if it’s not easily found



Selected user: Jeff Andrews