SIGMA

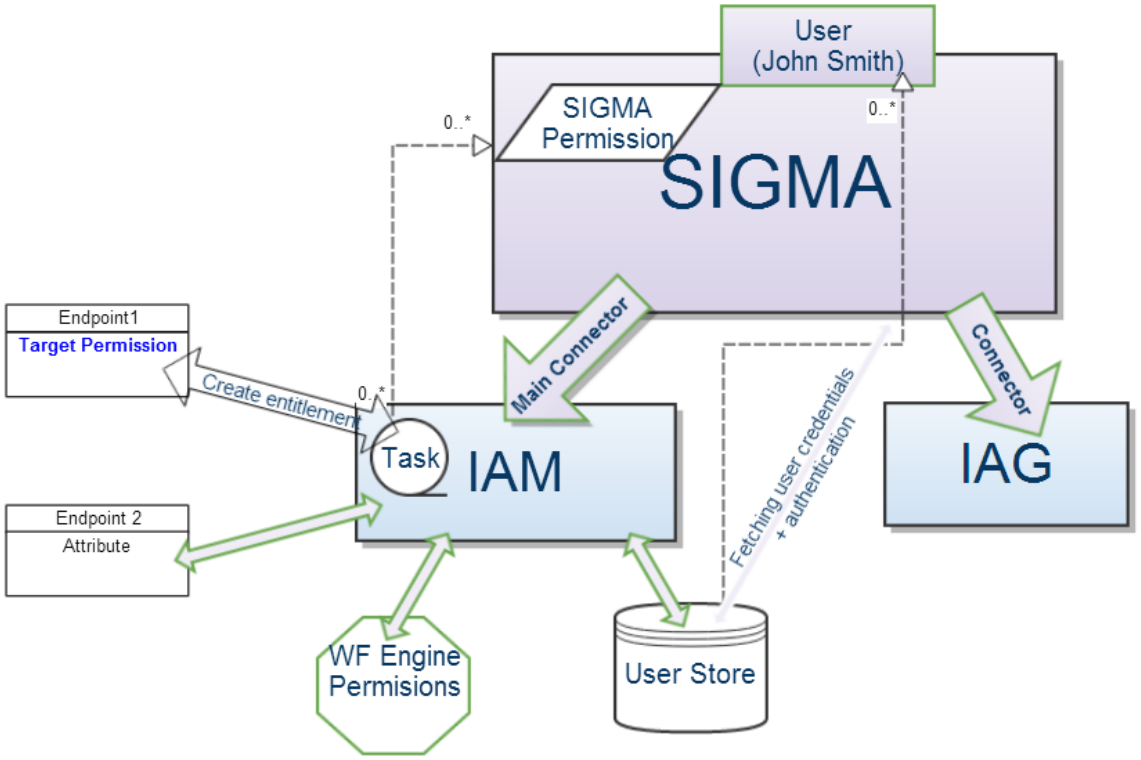
System Administrator Guide

TABLE OF CONTENTS

INTRODUCTION

SIGMA is a web-based business-ready identity and access management application, which serves as a business logic layer that leverages and aggregates functionality from existing identity management products, such as CA IdentityMinder (CA IDM) and GovernanceMinder (CA GM). SIGMA is designed for the non-technical business end-user and delivers an intuitive, all-inclusive interface in the form of a single page web application.

SIGMA interfaces with the organization's existing IDM platforms (such as CA IdentityMinder) through SIGMA's backend connectors, and communicates with the IDM backend platforms using the exposed public APIs of these backend systems (for example, Web Services (TEWS) & Workpoint APIs for CA IDM, and web services API for CA GM).



SYSTEM ADMINISTRATION

SIGMA's administration application allows system administrators to perform the following administrative tasks:

1. Configuring SIGMA in the organizational environment and linking it to various organizational applications (IAM - Identity and Access Management / IAG - Identity and Access Governance), see the following section.
2. Defining the business logic of the SIGMA system, and workflows derived from that logic.
3. Configuring the end-user interface based on these business logic definitions – see *User Configuration*, page 9

In order to perform administrator tasks in SIGMA you must be logged in with a system admin account

CONFIGURING SIGMA IN THE ORGANIZATIONAL ENVIRONMENT

Configuring SIGMA in the organizational environment consists of creating interfaces to the organization's IAM/IAG's by configuring SIGMA's connectors.

CONNECTORS

SIGMA uses connectors to communicate with IAM/IAG's.

The connectors enable SIGMA to perform the following tasks:

1. Authenticate/authorize users to SIGMA's interface.
2. Fetch exiting entitlements and expose them to end user.
3. Request entitlements.
4. Update statuses or ongoing activities.

SIGMA's factory settings support 3 types of connectors:

1. **IM Connector** – supports connectivity to the various IM versions.
2. **GM Connector** – supports connectivity to various GM versions.

For supported versions, please refer to the SIGMA's Platform Support Matrix.

3. **DB Connector** – A connector to a custom database which allows you to define your own entities and save their current state.

SIGMA also supplies an SDK for implementing your own custom connector. For more information please see the SIGMA's Programming Guide.

Configuring connectors is the first task that has to be completed in SIGMA's setup process. Complete all required connectors before proceeding with the rest of the configuration.

MAIN CONNECTOR

The Main Connector identifies a connector as the authoritative source that will be used by SIGMA for user authentication.

It is recommended that the Main Connector will be connected to the IAM/IAG system which contains the most extensive information of users in the organization.

CREATING A CONNECTOR

To create a connector you must have the IAM/IAG set-up. You will need to collect basic connectivity information on the endpoint to which you would like to connect before creating the connector. This information is typically available in the endpoint administrative management consul.

To create a connector:

1. Select the ***Connectors*** tab.

2. Choose the ***New Connector*** button.
3. Fill in a name that will identify the connector.
4. Select the type of connector from the list.
5. If no Main Connector is defined in the system, an option to configure this connector as the main connector will be available.
6. Fill in all the information required by the connector. An explanation of the purpose and samples of values is available next to each field.
7. Once completed click ***Save***.

Upon saving the connector for the first time, the connector will attempt to load. If an error occurs you will receive an error message. If no error message is displayed the connector is created successfully.

To modify the connector settings, click on the ***Edit*** button next to the connector.

EDITING AN EXISTING CONNECTOR

1. Switch to the Connectors tab.
2. Click on ***Edit*** next to the connector you wish to edit.
3. Edit the connector's settings.
4. Click save.

Upon saving the connector, the configuration will be saved but the connector will not attempt to load with the new configuration. To start the connector with the new configuration, click the ***Reload*** button to restart the connector.

Restarting the connector will cause it to be unavailable for the duration of the restart.

CA IM CONNECTOR CONFIGURATION

CA IM connectors are defined per environment. The following parameters are used when defining a connector:

General attributes

1. Name – identifying name for the connector.

CA IM

1. **CA IM UserId** – the identifying attribute of a user in the CA IM endpoint, which can be found in the User directory mappings in the management console.
2. **TEWS wsdl URL** – this is a mandatory attribute which contains the URL for the WSDL, which is generated for the connector environment. Typically http://<server_address>/iam/im/TEWS6/<environment_private_name>?WSDL

3. **Management Console Url** – environment management console URL. Usually http://<server_name>/iam/immange/ (make sure to include the closing "/")
4. **Management console user id** – UserName to access the protected IM management console.
5. **Management console password** – Password to access the protected IM management console.
6. **Environment Id** – environment id number, can be found in the environment configuration in the management console.
7. **Environment Name** - environment name, can be found in the environment configuration in the management console.
8. **Static Roles and Tasks XML** – used as a static override for environment roles and tasks xml file, should be used only in debug mode.
9. **CAIM Admin userid** – service admin user for identity management.
10. **CAIM Admin password** – service admin user password for identity management.
11. **No compile list** – used to remove tasks from WSDL compiled classes.
12. **TEWS client dir** – directory for saving compiled classes used for TEWS.
13. **Roles and tasks converter** – this will define the version of IM, used to encrypt the format of roles and tasks.
14. Workpoint client directory – the folder in which the workpoint client jars reside.

Forgotten Password

1. Forgotten Password Task – the task tag of the IM forgotten task.
2. Forgotten Password answer attribute – prefix of the answer attribute. Note this is only a prefix, so no need to define all the different answers attributes.
3. Forgotten Password question attribute - prefix of the question attribute. Note this is only a prefix, so no need to define all the different questions attributes.
4. Forgotten Password attribute – the password attribute in the forgotten password task.
5. Forgotten Password confirm attribute – the confirm password attribute, in the forgotten password task.

Tasks

1. **Default Search Task Tag** – SIGMA service task for searching user information.
2. **Login Task**– SIGMA service task used to verify login.
3. **User Status Attribute** – the attribute in the Login Task that is used to verify if the login status (active, disabled, password must change and etc).
4. **Scope Task** – SIGMA service task used to fetch the tasks in the authorized scope for this user in the system:
    a. **Scope Task group filed** – used to describe *scope task* group association field.

b. **Scope Task execution field** - used to describe *scope task* tasks field.

c. **Scope Task roles field** – used to describe *Scope task* role field.

5. **Admin task task** – SIGMA service tasks used to view admin task configuration.
6. **Approval task search task** – task used to resolve the workitem approval task.
7. **Task Status task** – task to be used for status check.
8. **Worklist task** – task to be used for viewing user's work items.
9. **Group search task** – SIGMA service task, used to view group objects.
10. **Group name attribute**– attribute used to identify group membership, can be found in the user directory mappings.

Workpoint

1. **Workpoint context class** – used to define Workpoint client context class. This information is relevant for defining the Application Server on which the workpoint resides. Information can be fetched from the workpoint designer configuration file. Possible values are : weblogic.jndi.WLInitialContextFactory For weblogic application server. org.jnp.interfaces.NamingContextFactory For Jboss. com.ibm.websphere.naming.WsnInitialContextFactory for websphere.
2. **Workpoint service url** – used to define the URL used by the workpoint client to connect to the workpoint server. When workpoint is installed on JBOSS, this is usually the hostname of the server. On weblogic this is usually t3://<server address>, when websphere this is usually iiop://<server address>:2809

It's important to configure FQDN resolution between the servers before defining this parameter.

3. **Workpoint Service User Id** – application server administrative user, if defined.
4. **Workpoint service user password** – application server administrative user password, if defined.
5. **Workpoint User Id** – the user to authenticate, usually not used.
6. **Workpoint User Password** – the password to authenticate the user, usually not used.
7. **Workpoint service method**– agent connection type, usually EJB.
8. **Workpoint DB** – workpoint DB name, by default WPDS.
9. **Workpoint Application Server** – application server brand on which workpoint is installed, JBOSS and WEBLOGIC are supported.
10. **Worklist date format** – date format. i.e: EEE MMM d HH:m:s z yyyy

CA GM CONNECTOR CONFIGURATION

General Configuration

1. Admin Name – the service username used to authenticate against the web service. Usually AD1\Eadmin is used.

2. Admin Password – the password used for the service username to authenticate to the web service
3. Certification resource display field – the display field for the resource element.
4. Certification user display field – the display field for the user element.
5. Master configuration – the name of the master configuration in the specified universe.
6. Model configuration – the name of the model configuration in the specified universe.
7. Server Name – the address of the GM server.
8. Server Port – the port of the GM server.
9. Universe name – the universe in which the connector is connecting to.
10. GM server version – the version of the GM server. Supported are 12.5.7, 12.6.0 and 12.6.1. prior to 12.5.7 use 12.5.7.

USER CONFIGURATION

User configuration consists of the following steps:

1. Configuring user Information
2. End-user GUI configuration

USERS

The ***user*** is the most fundamental entity in the SIGMA application. The user entity is a representation of an organizational entity as it exists in the various IAM/IAG systems connected to SIGMA.

SIGMA does not save organizational users information. Instead, it fetches the user information from the connected systems on demand.

The representation of the SIGMA user is defined by mapping of attributes in SIGMA to attributes in the IAM/IAG systems. To configure that mapping, use the User Info section in the admin UI.

CONFIGURING USER INFORMATION

The user information is derived from mappings the SIGMA user attributes to the IAM/IAG attributes.

You need to map all the user attributes that you intend to use in the SIGMA UI configuration and in SIGMA's business logic.

For example:

If a CA IM type connector exists in the system, the “First Name” attribute of the SIGMA user can be connected to the %FIRST_NAME% attribute in the CA IM connector. This means that once a user entity is used in SIGMA, the First Name will be fetched from the %FIRST_NAME% attribute of the CA IM connector from.

To configure these mappings:

1. Switch to the User Info tab. The configured attributes will be displayed.
2. To create a new attribute, click on the ***Add*** button. For each attribute you’ll need to supply a name for that attribute (the SIGMA attribute name), select the connector (from the list of system defined connectors) from which to fetch the information, and select the attribute in the connector to map the attribute to.
3. Click ***Save*** to commit the changes made.

The available attributes are derived from SIGMA service task you supplied for “Default Search Task Tag” in the connector configuration. To add more attributes simply modify the search screen of that task in IM. Attributes defined in the search results will be available for mappings, attributes defined in the search attributes will be available as searchable attributes.

In order to change the existing configuration, simply modify the attributes displayed on the screen and save.

SEARCH

SIGMA allows searching for users in the SIGMA system. This option is available in various modules such us:

1. Searching for a user to request access for.
2. Searching for another user when filling a form.
3. Searching for a similar user in order to compare entitlements.

The SIGMA search is a free text search. The search will look in a set of defined attributes for the keyword(s) entered by the user. To define these attributes check the “Searchable” checkbox next to the attribute in the User Info tab.

GUI CONFIGURATION

The GUI Configuration tab allows you to define the information displayed in the various screens throughout the SIGMA application. The following is configurable:

1. The presentation of user information in various places in the application. For example: display the “First Name” and “Last Name” in the search results of the Access Rights search.
2. The messages displayed to users in case no items are available. For example: in the Tasks module, if no pending approvals are awaiting, the user can be prompted with a configured message such as “No pending approval, have a nice day”
3. General UI configuration such as:
    a. Bulk file limit
    b. System unique attributes – used to identify the user from a file.

My Team (Subordinates)

Sigma comes with out of the box capability to display a predefined search results before searching for a user.

Those search results will already be displayed when the user enters a search module. This is a useful functionality since most business users tend to look for users from their own team. To configure this predefined search edit the property called “namedquery_<subordniates> in the GUI configuration. The Format of this property is - <Attribute To Search in>,<Value of Attribute from Logged In user>. For example if we want to configure the my team to be my subordinates. We would configure it to Manager,Userid that way the logged in user Userid will be searched in the Manger attribute.

HOW TO CONFIGURE GUI CONFIGURATION

1. Switch to the GUI Configuration tab.
2. Configure each of the parameters as desired.
3. Parameters which are not configured will be blank, and using default values is possible by clicking on use defaults.

The configurable parameters are displayed on the screen. The available User attributes are listed to the right under ***User Info***. By hovering over each of the fields the application context, which this parameter refers to, will be displayed.

To simplify the configuration you can type the "{" key and the system will display the available user attributes that are defined in the system.

In case you supplied a user attribute which is not defined, the system will display it in the list on the right under ***Not configured.*** In addition, you'll be promoted with a message of undefined attributes in the user Info tab.

TARGET PERMISSIONS

The target permissions are the corner stones on which the SIGMA permission model is constructed. It is the technical permission that the user requests, overlayered and simplified by the SIGMA permission model.

A Target Permission is the entitlement representation in the systems (i.e. IM, GM) that are connected to SIGMA. Use target permission either for fetching the entitlements the user currently has, or for granting new entitlements to the user. The supported entitlements are:

- Provisioning Role (IM)
- Group Membership (IM)
- Attribute(IM)
- Role (GM)
- Resource (GM)

When designing a SIGMA setup and implementation, one needs to plan and configure the relevant target permissions as detailed below.

ASSIGNING A TARGET PERMISSION

Target permissions can be assigned in 2 ways:

1. Directly through the native implementation of the connector:
   - **GM** – Through the API native method.
   - **IM** – triggering the corresponding event (similar to assigning a provisioning role in the Provisioning Roles tab)
2. Indirectly through a dedicated API.
   - CA IM – through executing a task which will be responsible to assigning that task.

CONFIGURING TARGET PERMISSIONS

1. Switch to the Target Permission Tab. The configured target permissions will be displayed.
2. Click ***New*** to add a new target permission.
3. Select the Connector which is associated with the target permission. The available target permissions for that connection will be displayed.
4. Upon selection, the type of the target permission will be automatically indicated.

If a target permission is of the attribute type, a value needs to be supplied.

5. Click ***Save*** to finish the configuration.

TASKS

In essence, tasks are IAM/IAG procedures, accessible through the connector's API. Each connector interacts with an external system using a public API. The API procedures are the way to execute business logic in that system. SIGMA defines the tasks as the repository of API calls that can be used in order to define the business logic. The task name will define the API function that needs to be triggered.

When building the implementation the administrator needs to define the various API procedures, which must to be defined in SIGMA in order to request various target permissions in the system.

For approval procedures refer to ***Error! Reference source not found.,*** page **Error! Bookmark not defined.**.

A task configuration comprises the following elements:

1. **Connector** – the connector where task is executed.
2. **Task Name** – The endpoint unique identifier for executing this task
3. Task Configuration – Bulk Task, Execution Plan (direct/indirect)
4. Form – configure form properties and mapping to task attributes.

CONFIGURING A TASK

1. Select the Tasks Tab. The configured tasks are displayed.
2. To create a new Task click on the ***New Task*** button.
3. Select the connector this task resides in.
4. Under ***Name*** select the task name. The tasks are available from a drop down list.
5. Complete the task configuration

- **Description**: a descriptive name for the task in Sigma.
- **IsBulkTask**: select if the task is a bulk task.
- **additionOperation/removalOperation** – for details see *Assigning a target permission* page 12.
  - Select ***directChange*** to use a direct action
  - Select ***execute Task*** for indirect action.

  Refer to *Target Permission* for more information.
- **ActionTaskMappin**g – used to describe the mappings between actions in a file and backend tasks.
- **BulkConf** – use to define the format of the CSV file and the condition which constructs the file.

ActionTaskMapping and **BulkConf** are applicable to bulk tasks only.

6. Configure the tasks form properties.

FORMS

Forms are used to collect more data from the user when performing an action. Forms can be used when:

1. Requesting access for a permission. Will be displayed when clicking to add/remove/modify a permission.
2. Performing a User Management action.
3. Onboarding a User(s).
4. Approving a task

Each form is constructed from a set of props. Props are basically a field in the form that could have different types such as:

1. Text
2. Password
3. Checkbox
4. Radio Buttons
5. Single Select
6. Drop Down
7. Date Picker
8. File Attachment
9. CSV
10. Multi User
11. User Selector
12. Multi Select List
13. Message

Each prop type has its own type of configuration.

The following configuration is common to all prop types:

1. Name – this is the label of the field.
2. Target Name – this is the attribute in the Task that the value will be pushed to.
3. Reference – used to attach a variable name to this prop, used for referencing it from other props using handlers.
4. Server Type – the type of processing that needs to be done on the value from the server. for example if it's a file, the content needs to be saved on a file system for further use. By default String will be used
5. Hidden – the prop will be present in the form but not displayed, this can be changed using prop handlers.
6. Read only – user cannot edit the prop.
7. Mandatory – user must fill this, prop is marked with a red astrik, will be validated upon submit.

Configuring a form:

1. In the administrative console switch to the form tab.
2. Click on New Form button.
3. Assign a name to the form describing it's functionality.
4. Select the task to which the form is linked to. The form prop will be linked to this task screen attributes.

5. Use the "*add Prop"* button to create new props.
6. When creating a new prop by click on it's name you can change it's label.
7. Select the prop type from the list.
8. Continue to configure the prop parameters.

This module support some advanced functionality:

1. To control the order of the props, drag and drop props.
2. Multiple tabs can be configured in the form using the add Tab button.
3. A preview is available on the right side.
4. Indication icons are available on the prop header to indicate if a code is configured in this form and if all mandatory fields are set.

FORM PROP HANDLERS

Handlers are javascript code that is configured to run on UI event. SIGMA support 3 types of events on which handlers can be configured:

1. OnChange – on each key type of the value of the prop this handler will be executed.
2. Validation – this will be performed on submit.
3. Initialization – this will be performed once the form is loaded. After the values are pulled from the task. Must return true or false.

When configuring the handler 2 objects are available – prop and api. Prop is this prop object allowing us to modify change and pull information from. Api is used to generic object that lives in the form that enables us to fetch information from other props and call server function (*refer to server functions for more information)*.

The Admin UI prop handler is equipped with a javascript editor for ensuring a more ease of use.

Use the following example for reference of how to configure an onChange handler

```
function onChange(api, prop) {
      if (prop.value.length > 10) {
            prop.value = '';
            api.getProp('dropdown').value = 'v3';
      }
}
```

SIGMA PERMISSION MODEL

When an identity management solution grows, organizing the structure of the entitlements becomes a challenge. To address that process, a flexible structure needs to be deployed, which will enable users to quickly and easily locate the entitlements they need.

The SIGMA Permission Model consists of the following entities:

- Application groups
- Applications

- Permissions
- Role Groups
- Roles

The basic entity is the permission entity. A permission is the business representation of the entitlement the user requests. Once a permission is requested, SIGMA translates this business representation to the technical entitlements – the target permissions.

The following rules define the permission model:

1. A permission can be linked to many target permissions. Example: The permission “Internet Access” can consist of several technical entitlements, such as DMZ access, Soft Token account and corporate LAN access.
2. A target permission can be linked to many permissions. Example: Active Directory Group membership, which can be a provisioning role in an IM solution and can be linked to several business permission such as Network Access, Security Admins etc.
3. Permission can be linked under another permission. In this case the permission will have a parent-son relationship. This relationship will ensure that a son permission cannot be granted without requesting/having the parent permission. This situation is common in profile-based applications. The basic access to the application is defined as the parent permission, while the specific profile/role in the application is defined as son permissions or sub-permissions.
4. Every permission must be linked to an application. A permission cannot be linked to more than one application.
5. Application can be linked to multiple permissions.
6. A group of applications contains one or more applications.
7. A SIGMA role is a group of permissions which define an organizational role.
8. A group of roles contain one or more roles.
9. There is no limit to the number of son permissions nesting in the permission model. In essence every son permission of a permission can have its own son permission and so on.
10. Permission can be grouped in a grouped in a group of permissions. Grouping permissions together means they are mutually exclusive (only one can be selected).

MANAGING THE PERMISSION MODEL

SIGMA allows the administrator to draw the permission model in the way it will be presented to user. Using drag&drop you can add each of the entities described above, and place them in the desired layout in the model.

1. Switch to Permission Tree tab.
2. The left panel contains the Group of Applications and Applications.
3. To create a new group of applications click on ***Create Group***. A new group will be displayed. Double-click on it to rename.

4. To create an application under the group of application, select the group and click “Create Application”.
5. To create permission, select the application in the left panel and click create permission in the middle pane.
6. To create a group of permission click on the application and click ***Create Group*** in the middle pane.
7. To add permissions to a group click and select the group name to add a permission under it.
8. To create nested permissions, either create a permission while selecting the parent permission, or create the permission and drag and drop it under the parent permission.
9. Dragging and dropping permissions is available all across the permission tree.
10. Click ***Save*** to commit these changes (unsaved changes will be displayed in red).

CONNECTING PERMISSION TO TARGET PERMISSION

1. Select the Permission Tree tab.
2. Select the permission you wish to connect.
3. Once a permission is selected the right panel will display the list of available target permission in the system. Check the target permission you wish to connect the permission to.
4. Click ***Save*** to commit these changes.

CONFIGURING PERMISSION PROPERTIES

SIGMA enables administrators to enrich the permission tree with additional information in order to provide end-users more information about the permissions. This is used to help end-users finding the correct entitlement they wish. The information will be displayed with a small Info icon next to the entity.

To configure this additional information, select the entity and click on the Properties tab in the right panel. Once completed save your settings.

UNDERSTANDING SCOPING AND RULES

Administrative roles are used in identity management for managing individual business requirements.
A role defines what operations can be performed by a user.

These operations define the ability of a user to acquire access (or requesting one) for different entitlement or business flows in the organization.

When a user logs in to SIGMA, this information is pulled by SIGMA connectors. SIGMA then calculates and translates this information and allows the user to request access or trigger flows only to what he’s allowed to.

This calculation is performed in several scenarios:

1. After selecting a user in the access module - SIGMA calculates which permissions the logged in user is allowed to request for the selected user.

2. In the onboarding module – SIGMA calculates what onboarding operations are allowed to the logged-in user.
3. After selecting a user in the user management module – SIGMA will calculate what user operations (Invocation operation of type USER) are within the user scope on the selected user.

RULES

Rules are configuration entities which are triggered when a target permission is selected. The triggered tasks can vary according to multiple parameters:

- **Expression** - Using the Requester and Target User(s) information to define the population of the rule.
- **Request Mode** – The state of the access rights module. SIGMA supports access rights for a single user, access rights for a bulk of users, access rights after onboarding a user, and access rights after onboarding a bulk of users.
- **Action Type** (Add/Remove/Modify)

A rule entity contains the various parameters above. When a permission is requested the rule can trigger tasks based this factors and predefined actions.

Example: A manager can add and remove “Network access” permissions for his employee using the “Add Network Access by manager” and “Remove Network access by manager” tasks respectively. His employee can request to add the Network Access permission using the “Request your manager network access” task.

To implement this logic two rules have to defined on the target permission which provides the network access.

Rule 1 – Manager acting on his employee. For this rule, the relationship between the requester (manager) and the target user (the employee) has to be defined. Usually these relationships are defined by an attribute in the subordinate's user profile indicating the manager. The resulting rule will look like this: "*user.Manager = Requester*" should be created.

Once the population of the rule is configured, the next step is to define which task to execute when the Add/Modify/Delete actions are selected. In this example the “Add Network Access by manager” task will be linked to the ***Add*** action, and “Remove Network access by manager” will be linked on the ***Remove*** action.

Rule 2 – Every employee can request the permission for himself. This is useful when the executed task is configured with a workflow, requiring an approval of a supervisor. To configure this rule define an expression which identifies the employee such as Requester[‘User Type’] = ‘Employee’. Then configure the ***Add*** operation with the “Request your manager network access” task.

MODULES ACTION

You can enable users to define different actions in other modules by configuring module actions. Activities of this type may include:

1. Managing Users – changing a user name, user’s manager etc.
2. Onboarding entities – used usually to onboard users, create groups.
3. Changing Password When expired.

Each of these actions are connected to forms and in turn to a task. These action are basically an execution of a task in the endpoint with its associated form. (Similar to admin Task)

CREATING A MODULE ACTION

1. In the administrative console switch to the *modules* tab.
2. Select the specific module in which you would like to create a new action.
3. Click on the *add* button.
4. Assign a name to that action, this is the display name.
5. Select the form you would like to be used for that action.
6. Category is optional, is not displayed.

TYPES OF MODULE ACTIONS

1. User Management – Actions will be used to manage user, should use this to perform action on an **existing** user that are not related to access request.
2. Self Onboarding – used for single user onboarding, will appear in the onboarding module under SINGLE. Category is not mandatory.
3. Bulk Onboarding – used for onboarding multiple users. Refer to *Bulk Onboarding* for how to configure a form and a task connected to this operation.
4. Password must Change – action that is used when password expired, usually requires to be linked to a form who has password and confirm password attributes as mandatory.
5. Registration – an onboarding action that is public and exposed to every person who has access to the registration url. Can be access via the login screen.

WORKING WITH TASKS IN CA IM CONNECTOR

Example 1

Assuming CA IM implementation is defined with a provisioning role named “Basic Active Directory account” which assigns provisions and creates an account in Active Directory.

This role can be requested in two ways:

1. Go to ***Modify User Admin Task,*** select the Provisioning Role tab, and select the ***Basic Active Directory Account***.
2. Use a “Self Modify” Admin Task called “Request Network Account” which displays a profile screen where the user needs to enter the reason he needs the requested access. The submission of this form triggers, a workflow for his manager and the AD admin. Following the completion of the workflow, a The PolicyXpress Policy triggers and assigns the Provisioning role.

Two tasks have to be configured in SIGMA to support both of these options:

1. “Modify User” task. This task will later be configured when the target permission will be requested by an admin.
2. “Request Network Account” task. This task will be used for self-request of the target permission by a user who is allowed to use the “Request Network Account” task.

APPROVAL FORM

In the sigma approve process, you can configure an approval form. Since the task that is used for approval is derived from the IM/GM system, you only need to configure the approval task in sigma and link a form to it.

Once an approval action occurs for the user, the form linked to the approval task will be displayed in the middle pane for the approver.

PROGRAMMING GUIDE

RULES EXPRESSIONS.

Rules are written in Rhino JavaScript language and are used to determine whether the operations are permitted. Using rules we can enforce another flexible level of scoping on who is allowed to request permissions.

Rule sample:

```
if (requester['FirstName'].get(0).equals(\"Super\"))
{ return true;
} else {
 java.lang.System.out.println(\"rule is incorrect\");
 return false; }
```

BULK CONFIGURATION

To configure a task to execute a bulk loader, the bulk file format needs to be defined. A bulk file is constructed using the following inputs:

1. List of operations, each operation is constructed:
    a. Target permission.
    b. Gui action
2. List of User Ids
3. List of Form Properties

BulkConf Example:

```
{
	"actionsMap": [{
		"operation": "ADD",
		"targetName": "TP 1",
		"targetValue": null,
		"targetType": "ROLE",
		"action": "add TP1"
	}],
	"actionsDef": [{
		"headers": ["%FIRST_NAME%"],
		"fields": [{FirstName}],
		"action": "add TP1"
	}]
}
```

AttributeMappings Example:

```
Add TP1.Modify User
```

GENERAL CONFIGURATION

The following General configuration is available:

1. File Upload Root – location for file attachments will be saved on the server. file attachment can be configured in the form.
2. Temp File lifetime – controlling the time files that were uploaded but not submitted, will be saved on the server.
3. Logging – sigma enables administrators to enable client log which tracks the user UI actions for debugging purposes. The following configuration can be used to enable this process:
    a. Logging users – the userids to enable this process on, use None to disable.
    b. Logging interval – the time in seconds in which logged actions will be sent for the user browser to the server to be saved.
    c. Logging Level – 1 to 4. 4 is lowest.
4. Logout Url – the url to redirect the user when he clicks on logout.
5. SSO User ID Header – the Siteminder header that is used for identifying the user.

TOOLS

Use the Tools module and the search request option to find which task session id was triggered in IM per sigma request. The TaskSessionId will be available under the “BackendRequestId” Attribute.